

Sztuczna inteligencja. Wykład wstępny

Paweł Rychlikowski

Instytut Informatyki UWr

1 marca 2019

- Wykład (dzisiaj i co tydzień)
- Pracownia/ćwiczenia (standardowy rytm 2+1)
- Egzamin (możliwe zwolnienia)
- Różne informacje na SKOS

Jaki język programowania?

Zapytanie: best computer language for ai

- Porada 1: Python, R, Lisp, Prolog, Java
- Porada 2: R, Python, Lisp, Java, Prolog
- Porada 3: Python, Java, Julia, Haskell, Lisp
- Porada 4: Python, C++, Lisp, Java, Prolog, Javascript, Haskell

Dlaczego tak (wybrane argumenty)?

1. **Python**: prosty, uniwersalny, wiele bibliotek (m.in. uczenie maszynowe – ML), używany w wielu miejscach do nauki AI
2. **R**: wsparcie dla ML
3. **C++, Java, Go, ...**: szybkość symulacji
4. **Lisp, Prolog**: dobre dopasowanie do **niektórych** zadań AI

- Stuart Russel, Peter Norvig, Artificial Intelligence. A Modern Approach. 3rd edition (w Internecie leży pdf)
- Fajna, ale 1100 stron.

Wykład na Stanfordzie

CS221: Artificial Intelligence: Principles and Techniques

Sztuczna inteligencja. Definicja

Definicja (krótka)

Zdolność komputera (programu) do wykonywania zadań powszechnie kojarzonych z zachowaniem inteligentnym (ludzi lub zwierząt).

Definicja długa

Spis treści wybranego podręcznika o Sztucznej inteligencji

Podział dziedziny na przedmioty w II

Definicja długa – spis treści

Rozwiązywanie problemów przez przeszukiwanie, rozwiązywanie więzów, przeszukiwanie z oponentem, logika w opisie świata, wnioskowanie w logice, planowanie, modelowanie niepewności, reprezentacja wiedzy, wnioskowanie przy niepewności, podejmowanie decyzji, uczenie się z przykładów, uczenie się ze wzmocnieniem, przetwarzanie języka naturalnego, rozpoznawanie wzorców w obrazach i dźwiękach, robotyka, procedural content generation.

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Podział dziedziny na przedmioty w II

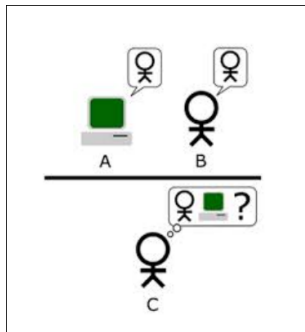
Definicja długa – spis treści

Rozwiązywanie problemów przez przeszukiwanie, rozwiązywanie więzów, przeszukiwanie z oponentem, logika w opisie świata, wnioskowanie w logice, planowanie, modelowanie niepewności, reprezentacja wiedzy, wnioskowanie przy niepewności, podejmowanie decyzji, uczenie się z przykładów, uczenie się ze wzmocnieniem, przetwarzanie języka naturalnego, rozpoznawanie wzorców w obrazach i dźwiękach, robotyka, procedural content generation.

Przedmioty:

1. Sztuczna inteligencja (nasz przedmiot)
2. Sieci neuronowe, eksploracja danych
3. Przetwarzanie języka naturalnego, Text mining, ...
4. Model checking, Sztuczna inteligencja i logika
5. Probabilistic graphical models
6. Sztuczna inteligencja w grach

Test Turinga



- Pytanie: Czy maszyny potrafią myśleć?
- Odpowiedź: Możemy to sprawdzić w rozmowie.

Pytanie

Czy **naprawdę** chcemy, żeby komputer był nierozróżnialny w rozmowie od człowieka?



źródło: tvtropes.org

Nieludzkie, pożądane cechy: wiedza, zdolności rachunkowe, modelowanie świata, rozwiązywanie zadań fizycznych, ...

- Przeszukiwanie (problem solving)
- Uczenie się (learning)
- Wnioskowanie (inference, theorem proving)
- Modelowanie wiedzy o świecie

Główny problem SI

Głównym, nierozwiązanym do tej pory problemem SI jest satysfakcjonujące połączenie tych światów, zwłaszcza wnioskowania (twarda logika), z rozmytą wiedzą pochodzącą z uczenia się.

Dlaczego problemy SI są trudne

Istnieją dwa główne powody

- 1 Trudność algorytmiczna:
 - **NP-zupełne** - łamigłówki,
 - **PSPACE-zupełne** - gry,
 - **EXPTIME i wyżej** – systemy wnioskowania
- 2 Czasem trudno sformalizować precyzyjnie problem!

Przykład

Rozważmy zadanie generowania tekstów w języku naturalnym (polskim, angielskim). Co to znaczy, że komputer robi to dobrze?

Co się udało Sztucznej inteligencji?

Gdzie komputery są od nas lepsze?

1. Oczywiście my też wykonujemy algorytmy (np. mnożenie liczb). Tu od zawsze **one** mają przewagę.
2. Komputery radzą sobie w grach: Szachy (1996), Go (2016), Poker (2017), StarCraft (2019?)
3. Rozpoznawanie mowy (Microsoft/IBM, Switchboard Corpora, około 2016)
4. Rozpoznawanie prostych obrazów (ImageNet, 1mln obrazków, 1000 klas):
 - 2010: około 28% błędów
 - 2015 (4.94% na ImageNet, człowiek: 5.1%)
 - (Obecny?) rekord: 3.5%
5. Tłumaczenie maszynowe (może niekoniecznie najbardziej skomplikowanych tekstów)
6. Wygrywanie teleturniejów wiedzy (Watson, Jeopardy,

Co działa niekoniecznie idealnie?

- 1 Rozpoznawanie mowy (50% błąd na nagraniach *przy koltlecie*)
- 2 Roboty umiejące nalewać wodę, otwierać drzwi, itd w **nieznanym** środowisku
 - DARPA challenge fails
- 3 Gra w Brydża
- 4 „Ludzka” rozmowa na dowolny temat

Paradoks Morav(e)ca

Paradoks, ok. 1980

Stosunkowo łatwo sprawić, żeby komputery przejawiały umiejętności dorosłego człowieka w testach na inteligencję albo w grze w warcaby, ale jest trudne albo wręcz niemożliwe zaprogramowanie im umiejętności rocznego dziecka w percepcji i mobilności. [Zgadza się?](#)

Steven Pinker

Gdy pojawi się nowa generacja inteligentnych urządzeń, to analitycy giełdowi, inżynierowie i ławnicy sądowi mogą zostać zastąpieni maszynami. Ogrodnicy, recepcjoniści i kucharze są bezpieczni w najbliższych dekadach

Łatwiej nam programować to co świadome (bo to lepiej rozumiemy), niż nieświadomość.

Sztucznie wygenerowany obraz, maksymalizujący **tosterowość**.



Co możemy zrobić z tym obrazkiem?

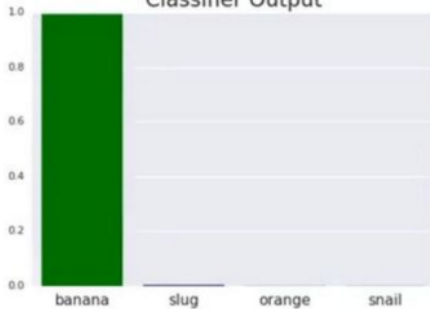
- Możemy go pokazywać sieci.
- Ale wklejając go analogowo, nie cyfrowo.
- Zobacz pracę: Adversarial Patch, T. Brown i inni, 2017

Tostery i banany

Classifier Input



Classifier Output

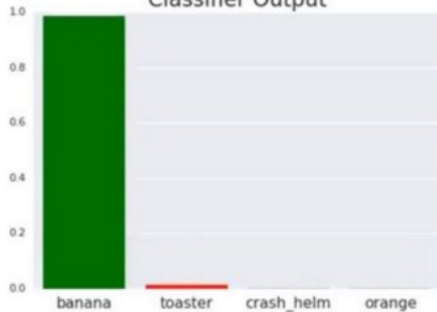


Tostery i banany

Classifier Input



Classifier Output

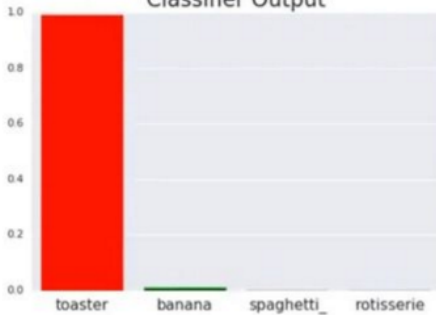


Tostery i banany

Classifier Input



Classifier Output



Najpierw zajmiemy się **przeszukiwaniem**, które jest podstawowym narzędziem AI.

Zadanie o Panu Tadeuszu

- a) **Przeszukujemy** możliwe podziały (sposoby legalnego wstawiania spacji)
- b) Można myśleć o tym jako o **modelu** języka: język składa się raczej z długich, niż z krótkich słów
- c) Uwaga: konieczny jest algorytm dynamiczny!

Zadanie o Pokerze

- a) Najpierw zagadka: Jaki jest związek tego zadania z pasjansem i bombą atomową?

- Stanisław Ulam pracował w projekcie Manhattan
- Kiedyś, w czasie rekonwalescencji po operacji układał pasjanse. A te wiadomo, czasem wychodzą, czasem nie.
- Chciał obliczyć prawdopodobieństwo (że pasjans wyjdzie), ale mu nie wychodziło

Wymyślił, że można je obliczyć symulując wiele rozdań na komputerze! (w roku 1947)

Tak powstały metody Monte Carlo (do wykorzystania w wielu miejscach, między innymi w zadaniu o pokerze).

Problem solving by searching. Intuicje

Przykład 1. Wyznaczanie trasy



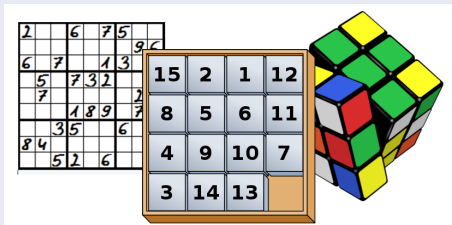
Przykład 2. Wyznaczanie sekwencji działań

Kohler (1945): monkey and banana problem.



Kohler observed that chimpanzees appeared to have an insight into the problem before solving it

Przykład 3. Rozwiązywanie łamigłówek



Uwaga

Problemy zabawkowe (toy problems) są częstym narzędziem w AI.

Czym jest problem?

Definicja

Problem ma następujące pięć komponentów

1. Stan początkowy (i zbiór stanów, ale być może dany implícite)
2. Zbiór akcji (co **agent** może robić)
3. Model przejścia ($\text{stan} + \text{akcja} = \text{nowy_stan}$)
4. Test określający, czy stan jest końcowy (i znaleźliśmy rozwiązanie)
5. Sposób obliczania kosztu ścieżki (najczęściej podawany jako koszt akcji w stanie)

Agent – program, który odbiera wrażenia o świecie, podejmuje decyzje, wykonuje akcje, maksymalizując jakąś funkcję celu.

Uwaga

Akcje oraz model przejścia są ze sobą powiązane. Jest kilka wariantów:

1. Zbiór akcji wspólny, funkcja przejścia pilnuje, żeby **niemożliwe** akcje nie zmieniały stanu
Przykład: **ludzik idzie w labiryncie na ścianę**
2. Akcje to funkcja, która dla stanu zwraca zbiór czynności, które agent może zrobić w stanie
3. Akcja i model przejścia to jedna funkcja, która dla stanu zwraca listę par postaci: (akcja, nowy_stan)

Zadania można modelować na wiele różnych sposobów. Dla **podróżowania MPK** mamy następujące akcje/sekwencje akcji

- Przejedź z przystanku A do przystanku B
- Wsiądź do tramwaju na przystanku A, skasuj bilet, zajmij wygodne miejsce, obserwuj tablicę, podejdź do drzwi, gdy...
- Wykonuj naprzemienne ruchy lewą i prawą nogą, aż ...
- Napnij głowę większą mięśnia dwugłowego lewego uda, ...

Uwaga

Akcje muszą być zrozumiałe dla agenta, im wyższy poziom, tym łatwiejsze zadanie przeszukiwania.

- Przeanalizujemy dla kilku przykładów jak można pewne zadania przedstawiać jako problemy przeszukiwania

Piętnastka

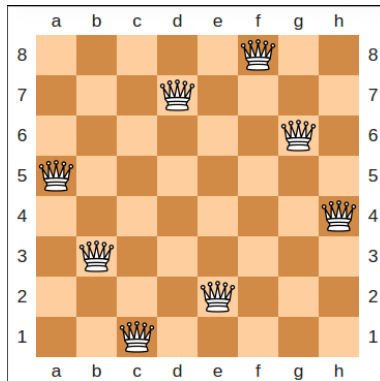


- **Stan początkowy:** jakieś ułożenie, na przykład powyższe
- **Stan końcowy:** liczby po kolei
- **Koszt:** jednostkowy
- **Model:** dowolna zamiana pustego kwadracika z sąsiadem

Zadanie szachowe z pierwszej listy

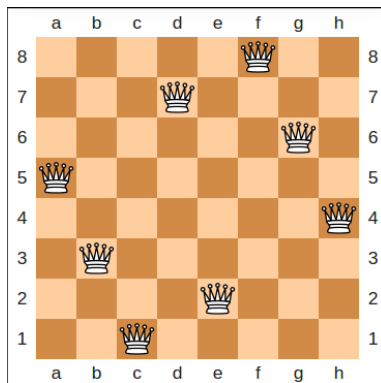
- **Stan początkowy**: jakieś ułożenie dwóch wież i króla, informacja o tym, kto się rusza
- **Stan końcowy**: mat
- **Koszt**: jednostkowy
- **Model**: ruch szachowy + zmiana gracza aktywnego

Problem ośmiu hetmanów



- W przeciwieństwie do poprzedniego przykładu istnieje tu wiele sformułowań.
- Jakie są dwie najważniejsze opcje?

Problem ośmiu hetmanów



Dwa sposoby opisywania zadania jako problemu przeszukiwania:

1. **Stan kompletny** (rozważamy sytuacje z 8 hetmanami na planszy, ruch to przestawienie hetmana)
2. **Stan niepełny** – zaczynamy od pustej planszy, ruchem jest postawienie hetmana.

Uwaga

Rozmieszczamy hetmany wg określonego schematu, rozpoczynając od pustej planszy. Liczymy liczbę stanów.

- $64 \times 63 \times \dots \times 57 = 178462987637760$ – rozmieszczamy po kolei hetmany, na jednym polu jest 1 hetman.
- $8^8 = 16777216$ – ustalona kolejność, w każdej kolumnie 1 hetman
- $8! = 40320$ – ustalona kolejność, w każdej kolumnie 1 hetman, nie szachują się w wierszach

Problem zabawkowy: hipoteza Knutha

Hipoteza

Zaczynając od 4 możemy dojść do dowolnej liczby wykonując operacje: **silni**, **pierwiastka** i **podłogi** (części całkowitej, `int`).

Przykładowo:

$$\lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \rfloor = 5$$

Sformułowanie dość oczywiste (stany to liczby)

Uwaga

Przestrzeń jest nieograniczona (nie znamy żadnego twierdzenia, które ograniczałoby przestrzeń prostą funkcją).

Modyfikacje zadania znajdowania marszruty

Przesuwamy się w stronę rzeczywistych zadań.

Wariant 1

W co K -tej miejscowości na trasie znajduje się dobra knajpa.

Wariant 2

Powinniśmy zaliczyć co najmniej K dodatkowych atrakcji turystycznych.

Zastanówmy się, co jest stanem w powyższych zadaniach?

Przykładowe zadania z rzeczywistego świata, będące zadaniami przeszukiwania:

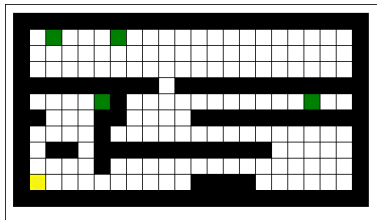
- Różne zagadnienia logistyczne
- Projektowanie układów scalonych
- Nawigacja robotów
- Organizacja procesu produkcji
- Tworzenie projektów o określonych właściwościach

- **Zupełność**: czy program znajdzie drogę do rozwiązania, jeżeli takowa istnieje?
Czy to jest konieczny warunek użyteczności algorytmu?
- **Optymalność**: czy będzie ona najkrótsza
- **Złożoność czasowa**: jak długo będzie trwało szukanie
- **Złożoność pamięciowa**: ile zużyjemy pamięci

Pytanie

Dlaczego tak ważna jest złożoność pamięciowa?

- Agent **bez pamięci** z konieczności operuje drzewem przeszukiwań (bo nie potrafi stwierdzić, że w jakimś stanie już był)
- Najczęściej lepiej modelować świat za pomocą **grafu**



- Będziemy teraz rozważać różne labirynty, na kwadratowej siatce.
- Labirynt jako problem wyszukiwania:
 - **stan** – współrzędne pola na którym można stanąć (nie ściany)
 - **start** – ustalona pozycja w labiryncie (żółta)
 - **cel** – ustalone pozycje w labiryncie (zielona)
 - **model** – 4-sąsiedztwo (modulo ściany), akcje to N, W, E, S.
 - **koszt** – jednostkowy