

Zagadnienia na zaliczenie - Elementów Robotyki Inteligentnej

1) Historia robotyki - ważne fakty mające wpływ na rozwój robotyki inteligentnej.

- Alan Turing w 1950r. „inteligentna maszyna”
- "Chiński pokój", John Searle w 1980r. „rozumiała maszyna”
- The Mathematics of Learning: Dealing with Data: Tomaso Poggio and Steve Smale - Systemy sztucznej inteligencji - które mając do zrealizowania złożony cel, poprzez pozyskiwanie i interpretowanie danych decydują o wyborze najlepszych akcji, do pewnego stopnia w sposób autonomiczny

Okres	Rozwój
1930-1950	(1930) Kurt Gödel: Udowodniono pewne twierdzenia dotyczące wnioskowania logicznego, znane jako 'twierdzenie o kompletności logiki pierwszego rzędu',
	(1936) Alan Turing: 'problemu stopu programu'
	(1950) Newell, Simon: (Na podstawie twierdzenia Goedle) pierwszy automatyczny system dowodzenia twierdzeń
	(1940) McCulloch, Pitts: Przedstawiono pierwszy model matematyczny dla sieci neuronowych
	(1950) Artykuł Turinga: Computing Machinery and Intelligence
1956	Spotkanie w Dartmouth: powstaje termin 'Sztuczna Inteligencja'
1960 - 1990	McCarthy: Wprowadzono LISP, język, który może przetwarzać struktury symboliczne
	(1965) Lotfi A. Zadeh: Zbiór rozmyte
	(1970) PROLOG, Język programowania został wprowadzony
	Wykazano, że sieci neuronowe są w stanie uczyć się na przykładach treningowych
	Nettalk, system, który był w stanie nauczyć się mowy z przykładowych tekstów (w oparciu o rozpoznawanie wzorców)
1990 - dotychczas	System hybrydowy łączący sieć neuronową i logikę rozmytą
	(1982) Teoria zbiorów przybliżonych, Z. Pawlak
	Eksploracja danych, Data science - ma na celu pozyskiwanie informacji z dużych zbiorów danych i bazy-wiedzy
	Rozproszone autonomiczne agenty, których celem jest rozwiązywanie problemów we współpracy wielu agentów

2) Test Turinga i jego kontr przykład – Chiński pokój, Czy test Turinga jest aktualny?

- Test Turinga – osoba A próbuje odróżnić maszynę od osoby B.
- "Chiński pokój" - osoba nie zna chińskiego, ma instrukcje, jak odpowiedzieć na pytania
- sam test może być niewystarczający jako jedyny wskaźnik inteligencji.

3) Słaba AI vs Silna AI (czym się różnią?; który wariant wystarczy do dorównania człowiekowi)?

- słaba SI / weak AI - zdolność do działania w sposób podobny do działania inteligencji człowieka, bez świadomości czy zrozumienia, sprecyzowane do konkretnych problemów, systemy rekomendacyjne, rozpoznawanie mowy
 - silna SI / strong AI - teoretyczny poziom zrozumienia i samoświadomości, porównywalny z ludzkim myśleniem, ogólne rozumienie świata i uczenie się, faktyczne myślenie w sposób niesymulowany.
-

4) Prawa robotyki Asimova

- 1) Robot nie może zranić człowieka ani przez zaniechanie działania dopuścić do jego nieszczęścia.
 - 2) Robot musi być posłuszny człowiekowi, chyba że stoi to w sprzeczności z Pierwszym Prawem.
 - 3) Robot musi dbać o siebie, o ile tylko nie stoi to w sprzeczności z Pierwszym lub Drugim Prawem.
 - 0) Robot nie może skrzywdzić ludzkości, lub poprzez zaniechanie działania doprowadzić do ich uszczerbku
-

5) Prawa robotyki w praktyce

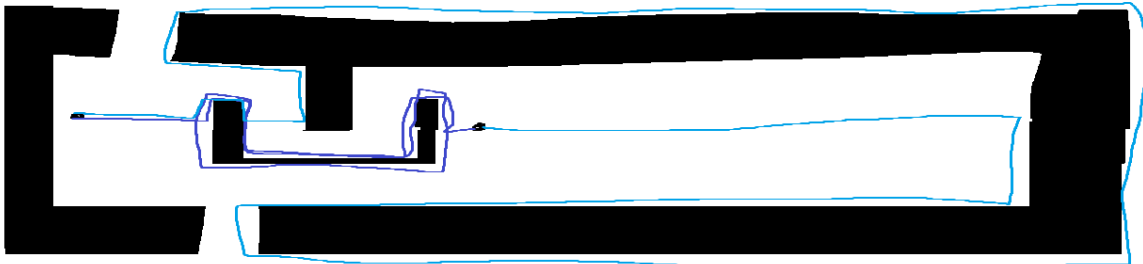
- Jeśli będą istnieć to powinny mieć takie same prawa i obowiązki
 - Instytucje, organizacje i eksperci pracują nad opracowywaniem:
 - Standardy etyczne i zasady projektowania systemów SI i robotów w celu zminimalizowanie ryzyka
 - Regulacje prawne i Odpowiedzialność projektantów i operatorów
 - Bezpieczeństwo i zgodność z prawem, Zintegrowane wartości etyczne wbudowane technologie.
-

6) Techniki lokalizacji (odnajdowania pozycji robota na mapie):

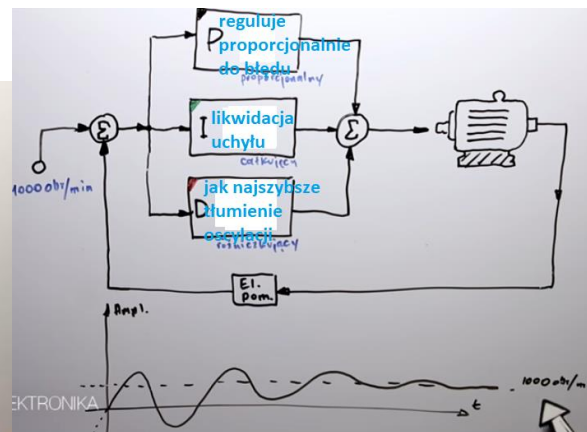
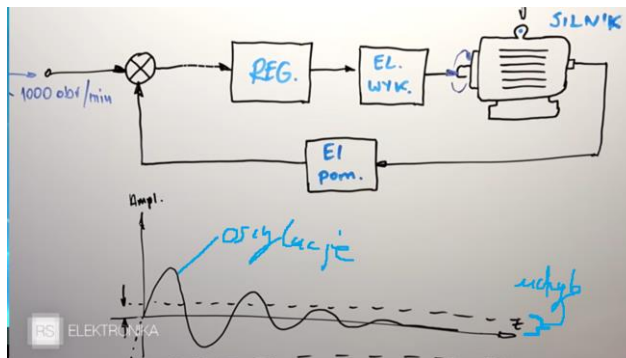
- a) filtr histogramowy / Bayesa - estymuje położenie obiektu w przestrzeni, często stosowany, zaletą jest możliwość uwzględnienia niepewności pomiarowej
- stworzenie mapy środowiska
 - nadanie każdej możliwej pozycji pewne początkowe prawdopodobieństwo
 - zebranie danych sensorycznych
 - aktualizacja prawdopodobieństwa
 - robot wykonuje ruch
- b) filtr partykułowy / metodzie Monte Carlo – korzysta z partykuł, reprezentacji potencjalnego położenia robota w przestrzeni, efektywny przy dużej niepewności i zmieniających się warunkach, elastyczny
- inicjalizacja partykuł
 - zebranie danych sensorycznych
 - aktualizacja wag partykuł
 - robot wykonuje ruch - każdy partykuła jest przemieszczany w sposób zgodny z przewidywanym ruchem
 - resampling – partykuły z mniejszymi wagami mogą być zastępowane przez kopie partykuł z większymi
- c) lokalizacja za pomocą znaczników mapy - określa położenie na podstawie znaczników w otoczeniu, użyteczny w środowiskach wewnętrznych, gdzie możliwe jest precyzyjne umieszczanie znaczników
- umieszczanie znaczników na mapie środowiska
 - odczytywanie znaczników za pomocą sensorów
 - porównywanie z mapą
 - system lokalizacyjny dokonuje aktualizacji położenia w miarę poruszania się robota
-

7) Planowanie trasy robotów mobilnych:

- a) BUG I – linia trasowa, obchodzi całe przeszkody, idzie do punktu najbliższego celu, nowa linia trasowa
- b) BUG II – linia trasowa, obchodzi przeszkodę aż dojdzie do linii trasowej, dalej idzie po niej
- c) Tangent BUG - linia trasowa, obchodzi przeszkodę aż stworzy nową linię trasową, nowa linia trasowa
- d) A^* w grafie – estymacja = koszt dojścia + heurystyka -> najmniejsza
- e) A^* w drzewie – estymacja = koszt dojścia + heurystyka -> najmniejsza
- f) A^* w siatce kwadratów pól (gridzie) - estymacja = koszt dojścia + heurystyka -> najmniejsza
- g) Technika generowania uniwersalnej polityki ruchu,
- h) Przykład, w którym BUG I generuje krótszą trasę niż BUG II.



8) Kontrolery / regulatory:



- P = proporcjonalność -> reguluje proporcjonalnie do błędu

- I = całkowanie -> likwidacja uchyłu

- D = różniczkowanie -> jak najszybsze tłumienie oscylacji

Jeśli robot porusza się wzdłuż prostej i zaczyna schodzić z kursu, regulator P zwiększy prędkość jednego z kół, aby skorygować błąd i przywrócić robot do pożądanej trajektorii.

9) Algorytm wygładzania ścieżki (Path smoothing) na bazie zejścia gradientowego.

- pomagają uzyskać bardziej efektywną i estetyczną trajektorię dla robotów mobilnych.

10) Znane mikrokontrolery mające zastosowanie w robotyce inteligentnej,

- Raspberry Pi:
 - Arduino:
 - NVIDIA Jetson:
 - STM32 (STM32F4, STM32H7 itp.):
 - ESP32:
 - BeagleBone:
 - Intel Edison/NUC:
 - PIC (PIC32):
-

11) Typy sensorów i ich możliwości – w kontekście Arduino Mega,

- czujniki odległości
 - czujniki światła i koloru
 - czujnik PIR, ruchu
 - czujniki dźwięku, wilgotności i temperatury, gazu, zbliżeniowe, Akcelerometry, Żyroskopy, dotyku, odcisków palców, Moduły GPS, Moduły Bluetooth, Kamery
-

12) Metodologia tworzenia produktów data miningowych przez pryzmat systemów interakcji Robot – Człowiek (Human – Robot interaction systems):

a) Kolejne etapy CRISP-DM - standard, struktury dla projektów związanych z eksploracją danych

+ techniki preprocesowania danych:

- normalizacja - Dostosowanie wartości różnych zmiennych do wspólnego zakresu [0, 1].
- standaryzacja - Przekształcenie tak, aby miały średnią wartość = 0 i odchylenie standardowe = 1.
- dyskretyzacja - Przekształcenie zmiennych ciągłych na kategorie poprzez podział na przedziały.
- zamiana na dummy variables - Przekształcenie zmiennych kategoriowych na binarne (0 lub 1
- techniki uzupełniania wartości nieznanymi (missing values) – medianą, średnią, zerem

+ metody oceny jakości modeli:

- Trenuj i Testuj (Train and Test) - Model jest trenowany na zbiorze treningowym i oceniany na testowym
- Walidacja Krzyżowa Monte – Carlo - Losowe podziały danych na zbiory treningowe i testowe w celu wielokrotnej walidacji.
- Walidacja Krzyżowa - Podział danych na k-krotne zestawy treningowe i testowe.
- Walidacja Krzyżowa wewnętrzna - Wewnętrzne k-krotne walidacje dla poszczególnych zbiorów danych.
- Wielokrotny Bootstrap - Losowy wybór próbek ze zwracaniem, aby uzyskać wiele zbiorów treningowych.
- Leave One Out - każdy punkt danych jest używany jako zestaw testowy, a reszta jako treningowy.

+ parametry szacowania jakości modeli:

- dokładność globalna (Accuracy globalna) - Mierzy odsetek poprawnie sklasyfikowanych obserwacji
- dokładność zbalansowana (balanced accuracy) - średnia arytmetyczna dokładności dla każdej z klas
- precision - poprawnie sklasyfikowanych pozytywnych przypadków / wszystkich przypadków
- recall - poprawnie sklasyfikowanych pozytywnych przypadków / rzeczywistych pozytywnych przypadków

- Krzywa precision-recall - graficzna reprezentacja zmiany precision i recall w zależności od progu decyzyjnego modelu
- Pokrycie (Coverage) - odsetek przypadków, dla których model przewidział pewność powyżej określonego progu
- + Przykłady wybranych technik klasyfikacji – ich idea
- Regresja Logistyczna - próg decyzyjny, aby dokonać klasyfikacji do jednej z dwóch klas
- Maszyna Wektorów Nośnych (SVM) - znalezienia hiperpłaszczyzny, która jak najlepiej separuje dane należące do różnych klas
- Drzewa Decyzyjne - podział zbioru danych na podstawie różnych cech
- + Przykłady wybranych technik wzmacniania efektywności modeli,
- Random Forests - Tworzy wiele drzew decyzyjnych podczas treningu i łączy ich prognozy, aby uzyskać bardziej stabilne i dokładne wyniki
- Komitet Bootstrapów (bagging) - Każdy model w komitecie jest trenowany na innym podzbiorze danych. Następnie prognozy z wszystkich modeli są łączone, co pomaga w redukcji wariancji modelu.
- AdaBoost (boosting) - przydziela wagi próbkom danych i dostosowuje je podczas kolejnych iteracji treningu
- + Przykład formowania finalnego produktu Data Miningowego
- Definicja celu biznesowego
- Zbieranie danych
- Eksploracyjna analiza danych
- Preprocessing danych
- Wybór modelu
- Trenowanie modelu
- Walidacja modelu
- Optymalizacja modelu
- Implementacja w środowisku produkcyjnym
- Monitorowanie i utrzymanie
- Dokumentacja