

Inżynieria Mechatroniczna

Podstawy Sztucznej Inteligencji i Uczenia Głębokiego:

1: Wprowadzenie, Optymalizacja

Ziemowit Dworakowski
AGH w Krakowie

1

Zanim zaczniemy...

Prezentacje wykładowe (i wszystkie pozostałe materiały z przedmiotu) będą dostępne na mojej stronie 2 dni przed danym wykładem lub laboratorium:

<http://galaxy.agh.edu.pl/~zdw/students.html>

Sugeruję robienie notatek tylko dotyczących rzeczy których NIE MA na slajdach. Podczas wykładu warto się skupić na zrozumieniu pojęć i toku rozumowania, łatwiej o zapamiętanie treści, gdy zrozumiecie ogólną ideę.

Przedmiot rusza po raz pierwszy – więc traktujemy go eksperymentalnie. Informacja zwrotna od Was jest kluczowa, jeśli coś nie idzie tak jak powinno, nie czekajcie do końca przedmiotu, dajcie znać od razu

2






Dr hab. inż. Ziemowit Dworakowski, zdw@agh.edu.pl
(koordynator kursu na IME)
Konsultacje: D3, 4.17, Czwartek, 8:00 – 10:00
Zakres ekspertyzy:
AI, systemy decyzyjne, optymalizacja, przetwarzanie sygnałów i obrazów, systemy wizyjne

Dr inż. Krzysztof Holak, holak@agh.edu.pl
(koordynator kursu na IME)
Konsultacje: D3, 4.04, Czwartek, 9:00 – 11:00
Zakres ekspertyzy:
systemy wizyjne, przetwarzanie sygnałów i obrazów, inżynieria mechaniczna

Mgr inż. Adam Machynia, machynia@agh.edu.pl
Konsultacje: D3, 4.08, Czwartek, 10:00 AM – 11:30 AM
Zakres ekspertyzy:
AI, przetwarzanie obrazów, przetwarzanie sygnałów

Mgr inż. Jakub Szczepaniak, jakub@agh.edu.pl
Konsultacje: D3, 3.08, Thursday, 10:00 AM – 11:30 AM
Zakres ekspertyzy:
AI, monitorowanie samolotów, przetwarzanie sygnałów

Course team

Prosimy o powiadomienie mailowe odpowiedniego prowadzącego przed konsultacjami

3

Po co jest ten przedmiot?

Pokażemy Wam, jak działają „metody AI”

Pokażemy Wam, jak „metody AI” działają w inżynierii

Nauczmy Was przetwarzania rzeczywistych zbiorów danych (w tym sygnałów czasowych i obrazów) z zastosowaniem uczenia maszynowego i AI

Pokażemy Wam nieco z aktualnego stanu wiedzy

Będziecie w stanie podjąć decyzję, czy temat jest interesujący i czy chcecie się nim zająć w przyszłości

Nauczmy Was sposobu myślenia, który pozwoli zrozumieć metody AI (również te dopiero powstające) ... i projektować swoje własne systemy decyzyjne

4

Jak zorganizowany jest przedmiot?

Mamy:

14 wykładów
13 laboratoriów

Optymalizacja

Uczenie maszynowe

Przetwarzanie obrazów

Metaparametry

Uczenie głębokie i zaawansowane AI

★ Testy
★ Poprawa testów / duży test
★ Odpowiedzi na pytania (możliwe zawsze, choć nieczęste)

5

OK, ale jak dostać zaliczenie?

- Trzeba zaliczyć wszystkie instrukcje (każda kończy się oceną)
- Trzeba zaliczyć wszystkie testy

Ocena końcowa = średnia wszystkich ocen

6

Laboratoria wykorzystują **odwroconą klasę**:

- Przygotowujecie się do zajęć w domu (Sugeruję wykonanie zadań oznaczonych „na 3.0”)
- Im więcej zadań ukończycie na laboratoriach – tym wyższą ocenę otrzymujecie
- Nie będzie konieczności pisania sprawozdań, pod warunkiem że wykonacie co najmniej zadania „na 3.0” w trakcie zajęć **oraz** zaprezentujecie wszystkie zadania „na 4.0” i „na 5.0” na **następnych** zajęciach.

(Szczegóły znajdują się w pierwszej instrukcji laboratoryjnej)

7

Co w przypadku nieprzewidzianych trudności?

Brak zaliczenia laboratorium (z powodu słabego przygotowania lub nieobecności) wymaga przygotowania standardowego raportu z kompletu zadań + jednego dodatkowego (wybranego przez nauczyciela). Można tak zrobić do trzech razy podczas semestru.

Na koniec semestru będzie możliwość poprawy dowolnych trzech (z siedmiu) testów. Więcej braków będzie wymagało zaliczenia kolokwium z kompletu materiału z zajęć.

8

Czy w tym pokoju znajduje się stół?

Tzn. co z ChatemGPT (i wszystkimi innymi narzędziami GenAI)?

Co jest konieczne:

- Informacja o sposobie użycia AI – jeśli jej używacie (np. „raport przygotowany z zastosowaniem ChatGPT w celu...”)
- Branie odpowiedzialności za swoją pracę (tzn. jeśli używacie GenAI np. do przygotowania kodów, nadal musicie je rozumieć i być w stanie wyjaśnić – jakby były Wasze)

Czego nie można robić:

- Używać GenAI w jakiegokolwiek formie podczas zajęć, chyba, że nauczyciel wyraźnie wyrazi na to zgodę
- Używać GenAI do pisania i przepisywania kodów

Co jest OK:

- Używanie GenAI do poprawiania błędów w kodach
- Używanie GenAI do przygotowywania fragmentów raportów z promptów – przy czym prompty należy zachować do wglądu na prośbę nauczyciela
- Używanie GenAI do poprawy pisowni

Dlaczego?

Ponieważ instrukcje laboratoryjne już teraz zapewniają prawie-kompletne kody – o celem zajęć jest nauczenie Was o co w nich chodzi i jak ich używać.

9

Optymalizacja

10

Rzeczywisty problem



- Jak zarządzać linią lotniczą by zmaksymalizować zysk?
- Jak zaprojektować samochód, by był wytrzymały, ale tani i cichy?
- Jak spędzić czas na studiach w najefektywniejszy sposób?

Opis
problemu

Parametryzacja

Parametry

Informują: czym możemy „sterować”,
na co mamy wpływ?

Które wartości parametrów
są „najlepsze”?

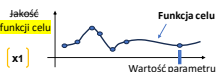
Optymalizacja oznacza poszukiwanie
globalnego optimum funkcji celu
zdefiniowanej w przestrzeni parametrów

Optymalizacja

11

Jak odpowiadać na takie pytania?

Uprośćmy do wyłącznie
jednego parametru:

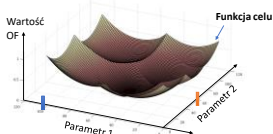


To jakoby przesuwac suwak,
szukając najlepszego położenia...

Jeśli parametry są
dwa...

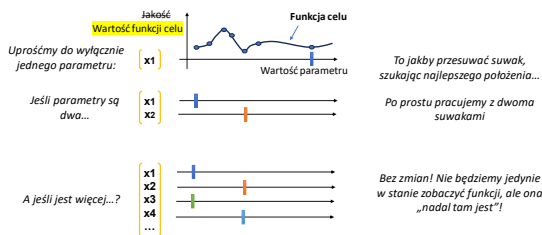


Po prostu pracujemy z dwoma
suwakami



12

Jak odpowiadać na takie pytania?



13

Jak odpowiadać na takie pytania?

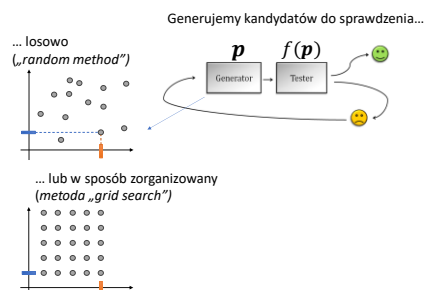
$$f(p) \rightarrow \frac{df(p)}{dp}$$

Poszukujemy ekstremum, gdzie pochodna się zeruje...

W typowych problemach nie znamy równania funkcji. Zatem jedynym sposobem na rozwiązanie problemu jest podejście metodą „prób i błędów”

14

Jak odpowiadać na takie pytania?



15

Jak odpowiadać na takie pytania?

... losowo
(„random method”)



Działa dobrze, pod warunkiem, że nie ma zbyt wielu parametrów (Tylko kilka „suwaków”)

Zupełnie nie działa jeśli ilość parametrów jest duża (mniej więcej: większa niż 3)

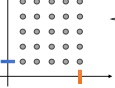
Tutaj wykonaliśmy 25 „prób” (5*5)

3D -> 125

4D -> 625

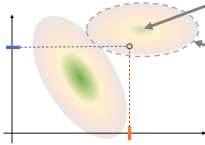
10D -> 9 765 625 prób...

... lub w sposób zorganizowany
(metoda „grid search”)



16

Metoda 1+1



Taki punkt nazywany jest **lokalnym minimum funkcji celu** (to punkt, który ma najniższą wartość w swoim niezerowym otoczeniu)

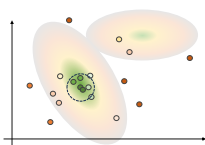
Ale jeśli zaczniemy optymalizację wewnątrz **obszaru przyciągania minimum lokalnego**, nie znajdziemy minimum globalnego!

Jak sobie poradzić z tym problemem?

17

Metoda 1+1

Algorytm:



Zacznij od losowo wygenerowanego kandydata p_c

Sprawdź jego wartość funkcji celu: $f(p_c)$

Jeśli jest jak do tej pory najlepsza, zachowaj ją: $p_b = p_c$

Wygeneruj nowego kandydata w sąsiedztwie zachowanego punktu: $p_c = p_b + S \cdot r$
(S to wielkość sąsiedztwa, r to wektor losowy)

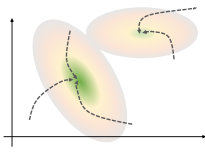
* Jeśli progres się zatrzymał, zmniejsz S

* Jeśli S jest bardzo małe i progres się zatrzymał, zakończ optymalizację

- Dobra skalowalność w średnio-wysoką wymiarowość przestrzeni parametrów
- Relatywnie łatwa w konfiguracji
- Działa również w przypadku nieciągłej przestrzeni parametrów

18

Metoda gradientowa



Koncepcja:

Zaczynamy ponownie od dowolnego punktu
 ↓
 Sprawdzamy „teren dookoła” by ocenić gdzie funkcja opada najszybciej
 ↓
 Wykonujemy duży krok w kierunku największego spadku

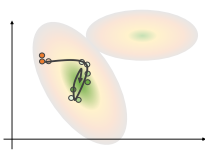
Będziemy podążać w stronę minimum przez cały czas – ale możemy też wokół niego oscylować...
 Więc ponownie możemy zredukować wielkość kroku w przypadku braku postępów!

Możemy również poprawić metodę poprzez dodanie bezwładności (**momentum**) – by pozwolić jej „nabrać prędkości”

A ponieważ metoda również będzie przyciągana przez minima lokalne, możemy uruchomić ją kilkakrotnie (**multistart**)

19

Metoda gradientowa



Algorytm:

→ Zaczynamy od losowo wygenerowanego kandydata p_c

↓
 Wyznacz gradient funkcji celu wokół kandydata:
 $\nabla f(p_c)$

↓
 Wykonaj krok w kierunku największego spadku gradientu: $p_n = p_c - \epsilon \cdot \nabla f(p_c)$
 Ten punkt staje się nowym kandydatem: $p_c := p_n$

- W przypadku braku progressu zmniejsz ϵ
 - Powtarzaj określoną ilość iteracji lub aż do wyzerowania gradientu

↓
 (możesz rozpocząć metodę kilkakrotnie, zapisując najlepszy rezultat)

- Dobra skalowalność w przestrzenie wysoko-wymiarowe
- Łatwa w konfiguracji
- Wymaga ciągłej (różniczkowalnej) funkcji celu

20

Porada dnia

Wydałoby się, że metoda gradientowa powinna być ZAWSZE uruchamiana w wersji wielostartowej. Ale okazuje się, że to nieprawda. W **wielu** praktycznych problemach optymalizacyjnych, im więcej parametrów mamy do dyspozycji, tym większa szansa, że minimum jest w rzeczywistości tylko jedno (lub jest ich wiele, ale wszystkie mają podobną głębokość) ...

Wyobraźmy sobie grupę ludzi wykonujących wspólnie duży projekt:



Im więcej ludzi, tym łatwiej rozdzielić pomiędzy nich zadania!

21

Reprezentacje wiedzy

22

Reprezentacja 1:

Wektor proporcji składników



x1 – ilość wody [litry]
x2 – ilość maki [gramy]
x3 – drożdże [gramy]
x4 – nasiona [gramy]
...



Mamy tu konfiguracyjną decyzję do podjęcia:
Im bardziej złożona i elastyczna reprezentacja, tym potencjalnie lepszy rezultat końcowy.
Ale odnalezienie go może być dużym wyzwaniem...

Łatwiej tutaj zbudować „solver” i szybko dotrzemy do minimum. Ale minimum w tej uproszczonej reprezentacji wcale nie musi być optimum globalnym...

Reprezentacja 2:

Proces, z opisem działań i wejść



x1 – Akcja: dodać mąkę
x2 – Objętość: k gramów
x3 – Akcja: dodać drożdże
x4 – Objętość: k gramów
...



Mamy tu bardzo wiele złożonych parametrów. Proces optymalizacji może się zapętlić, być przyciągany przez minima lokalne... Być może nie osiągniemy w ogóle żadnego minimum...

23

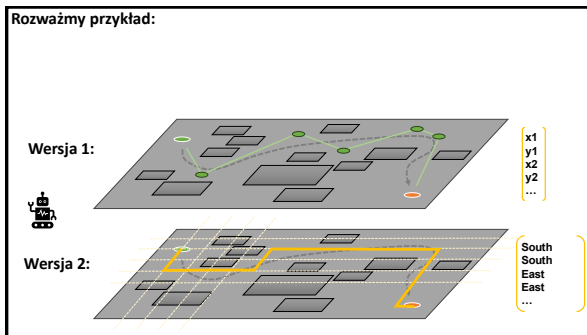
Projekt reprezentacji w praktyce:

Warto rozważyć:

- **Nad czym właściwie mamy kontrolę?**
(Nie zajmujemy się zmiennymi, których nie można kontrolować)
- **Co nam mówi ekspercka wiedza kontekstowa?**
(W większości typowych zastosowań można znaleźć kogoś, kto potrafi przynajmniej zasugerować właściwy tok postępowania)
- **Jak dużo czasu i zasobów jesteśmy w stanie przeznaczyć na zadanie?**
(Im mniej danych (pomiarów, symulacji) i czasu, tym prostsza musi być reprezentacja)
- **Jak trudno wykonać test wartości funkcji celu?**
(czyli: ile czasu zajmuje przejście od wartości parametrów do uzyskania odpowiadającej im wartości funkcji celu?)

24

Rozważmy przykład:



25

Funkcja celu może być:

- Jednokryterialna lub wielokryterialna



To oznacza, że mamy więcej niż jeden cel optymalizacji
 (np.: zaprojektować samochód tak tani, jak to możliwe,
 ale tak bezpieczny, jak to możliwe)

1) Możemy skleić wszystkie kryteria
 w jedno używając średniej ważonej

$$Q = W_1 \cdot \frac{1}{C} + W_2 \cdot \frac{1}{S}$$



2) Możemy zastosować
 Front Pareto

Tzn. szukamy zbioru różnych rozwiązań
 a potem porównujemy je ze sobą
 odrzucając obiektywnie gorsze (pod
 względem wszystkich kryteriów)

26

Funkcja celu może być:

- Jednokryterialna lub wielokryterialna
 - Ciągła (różniczkowalna) lub dyskretna



To oznacza, że wartości parametrów mogą być zmieniane z
 dowolną dokładnością (np. możemy zmienić szerokość karoserii
 samochodu z dowolną dokładnością, ale nie możemy dodać 2/3
 fotela do auta)

Dyskretna przestrzeń parametrów oznacza, że nie możemy
 zastosować metody gradientowej

27

Funkcja celu może być:

- Jednokryterialna lub wielokryterialna
- Ciągła (różniczkowalna) lub dyskretna
- Z ograniczeniami lub bez ograniczeń



To oznacza, że wszystkie wartości parametrów są dopuszczalne



To oznacza, że istnieją pewne dodatkowe warunki, które trzeba spełnić – np. projektując trajektorię nie możemy legalnie znaleźć się na autostradzie na rowerze lub na chodniku jadąc samochodem

Możemy albo zbudować reprezentację uniemożliwiającą wybór „nielegalnych” rozwiązań albo „ulegalniać” je już po odnalezieniu

28

Powtórka:

1. Podaj definicję optymalizacji i jej praktyczne przykłady (w tym: optymalizację z ograniczeniami, wielokryterialną, dyskretną lub ciągłą, itd.)
2. Wyjaśnij dlaczego wybór reprezentacji i poprawna parametryzacja są istotne – i jakie ryzyka są związane z tym krokiem
3. Wyjaśnij zasadę działania algorytmów losowego i *grid search*
4. Wyjaśnij zasadę działania, zalety, podaj algorytm i zilustruj algorytm 1+1
5. Wyjaśnij zasadę działania, zalety, podaj algorytm i zilustruj algorytm gradientowy
6. Czym są minima lokalne i jak możemy sobie z nimi poradzić w optymalizacji?

29
