

## PROJEKT 2

# ROZWIĄZYWANIE PROBLEMÓW OPTYMALIZACYJNYCH, STEROWANIE AGENTAMI, ANALIZA I GENERACJA TEKSTU LUB SZEREGÓW CZASOWYCH

### CEL I TEMATYKA PROJEKTU

Celem projektu jest wykorzystanie technik poznanych na zajęciach w drugiej części semestru w większym, samodzielnym zadaniu lub rozwinięcie pierwszego projektu o nowopoznane techniki. W szczególności chodzi o następujące techniki/algorytmy/modele:

- Algorytmy analizy tekstu (bag of words, wyciąganie wydźwięku / opinii, tematu z tekstu)
- Rekurencyjne sieci neuronowe (RNN, LSTM) oraz transformery lub inne modele deep learning
- Algorytmy genetyczne
- Inteligencja roju (PSO, ACO)
- Kontroler rozmyty
- Algorytmy Reinforcement Learning (Q-Learning, Deep-Q-Network, etc.)

Biorąc pod uwagę powyższe algorytmy, wydaje się, że wyłaniają się cztery główne pomysły na projekt nr 2:

#### 1) ROZWIĄZYWANIE PROBLEMU OPTYMALIZACYJNEGO

Bierzemy problem do rozwiązania np. wspomniany na wykładzie nonogram/obrazek logiczny, albo jakiś graf, a następnie szukamy rozwiązania tego problemu. Projekt polegałby na tym, że testujemy skuteczność paru algorytmów i porównujemy ich skuteczność (czy zawsze rozwiązują problem, niezależnie od jego wielkości?) oraz czas (czy wykonują się szybko?). W zależności od problemu warto porównać różne wersje algorytmu genetycznego (np. różne typy kodowania chromosomów, różne funkcje fitness), algorytm PSO lub ACO, a nawet (jeśli to możliwe) algorytmy RL. Można do tego dać jakiś bazowy algorytm typu brute force. Dobrze by było porównać 3 do 5 algorytmów dla kilku wersji problemu o różnym stopniu skomplikowania. Podsumowaniem takiego projektu mogłaby być taka tabelka z wynikami, np.:

	Przeszukiwanie grafu wszcz	Algorytm genetyczny (Fitness 1)	Algorytm genetyczny (Fitness 2)	ACO	Q-Learning
Problem 5x5	100% (1s)	100% (10s)	100% (8s)	100% (15s)	100% (20s)
Problem 10x10	100% (5s)	90% (15s)	100% (10s)	90% (20s)	100% (30s)
Problem 15x15	100% (70s)	50% (30s)	90% (15s)	60% (30s)	100% (60s)
Problem 20x20	0% (--) timeout	0% (--)	20% (60s)	10% (50s)	90% (200s)

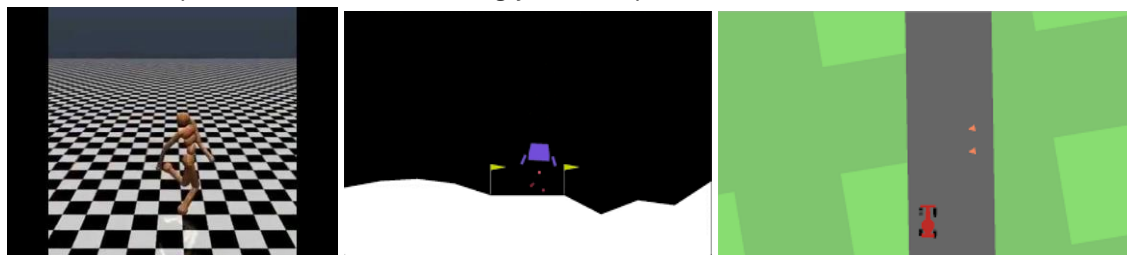
Takie liczby powinny być średnią wielu odpaleń algorytmu (minimum 10, lepiej więcej). W przypadku gdy algorytm znalazł rozwiązanie, można zapisać jego czas działania. Można do tego zrobić też fajne wykresy z wielkością problemu na osi X (5, 10, 15, 20) i skutecznością algorytmów na osi Y (linie w różnych kolorach dla różnych algorytmów).

Pomysły na problemy do rozwiązanie są niżej w sekcji Inspiracje tematów.

#### 2) STEROWANIE AGENTEM W WIRTUALNYM (LUB PRAWDZIWYM) ŚRODOWISKU

Celem projektu jest opracowanie algorytmu, który będzie dobrze sterował jakimś obiektem w danym środowisku (wirtualnym lub prawdziwym). Ponieważ nie mamy do dyspozycji robotów, dronów i innych hardwarowych kreatur, to musimy ograniczyć się do środowiska wirtualnego. Przykładem jest tu pokazana na wykładzie gra w czołgi, jak i inne gry. Można zresztą grę napisać samodzielnie i wytrenować do niej różne algorytmy. Są jednak środowiska oferujące gotowe gry lub wirtualne roboty. Przykładem jest Gym / Gymnasium

(<https://gymnasium.farama.org/index.html>) oferujące przeróżne problemy do rozwiązania: sterowanie samochodem, poruszanie robotem, różne gry arcade itp.



Gymnasium ma tę fajną cechę, że może odpalać grę wielokrotnie, trenując przy tym algorytmy. Można też wyłączyć interfejs graficzny, co przyspieszy proces uczenia.

W projekcie należałoby przetestować kilka algorytmów i porównać ich skuteczność, tj. sprawdzić jak dobrze sterują obiektem. Jeśli jest możliwość wystawienia oceny takim algorytmom (np. w ilu przypadkach doszły do celu gry, albo jak wysoko podniósł się wstający robot, albo jak daleko zaszliśmy w space shooterze) to warto zestawić te oceny w tabelce tak jak powyżej w pomyśle (1).

Jakie algorytmy warto przetestować:

- **Swój własny skrypt.** Można zapomnieć o AI i napisać prosty program do sterowania obiektem. Czasami proste rozwiązania są najlepsze. Oczywiście pytanie jest takie: czy nasz skrypt będzie na tyle elastyczny, że zadziała w każdej sytuacji i odnajdzie się w zmieniającym się środowisku?
- **Algorytmy RL.** Uczenie przez wzmacnianie, i nagradzanie agenta za wykonywanie dobrych akcji jest jedną z najpotężniejszych technik do stosowania w tym typie problemu. W zależności od typu gry i danych warto rozważyć Q-Learning lub Deep-Q-Network.
- **Kontroler rozmyty.** To wyzwanie innego gatunku: zamiast uczyć obiekt, opracowujemy rozmyty model, który ma szansę dobrze sterować agentem. Jeśli fuzzy controler umie sterować klimatyzacją czy parkować samochód, to może też będzie umiał wylądować sondą na księżycu (Lunar Lander). Taki rozmyty kontroler trzeba dobrze przemyśleć i pewnie wielokrotnie poprawiać. Warto dodać go jako technikę, zwłaszcza do problemów w których liczba zmiennych nie jest zbyt duża. Do sterowania kilkunastoma przegubami robota pewnie już będzie ta metoda za trudna do implementacji.
- **Algorytm genetyczny lub inteligencja roju.** Sterowanie obiektem można potraktować jako problem optymalizacyjny: szukamy jak najlepszego zestawu akcji, który gwarantuje dobre wykonanie akcji (np. wstanie przewróconego robota, wylądowanie sondy). Warto rozważyć, którąś z tych dwóch bio-inspirowanych metod metaheurystycznych.
- **Neuroewolucja (NEAT).** Jest technika, która wykracza poza materiał przedmiotu ale też może być bardzo ciekawa. Łączymy algorytmy genetyczne z sieciami neuronowymi w tzw. algorytmie NEAT. Algorytm genetyczny tworzy populację sieci neuronowych o różnych topologiach, a następnie ocenia ich fitness poprzez używanie ich w rozwiązywaniu problemu (np. graniu w grę).

Wybierając grę/wirtualne środowisko z Gymnasium, zwróć uwagę na to jakimi danymi opisane jest środowisko, oraz jakie akcje może wykonywać obiekt. Dane dyskretne/skończone z reguły są łatwiejsze niż zmiennoprzecinkowe.

### 3) ANALIZA TEKSTU Z MEDIÓW SPOŁECZNOŚCIOWYCH

Ciekawym projektem jest analiza wypowiedzi użytkowników mediów społecznościowych (Twitter/X, Facebook, Reddit, Telegram). W projekcie trzeba by zebrać dużą bazę danych postów (najlepiej powyżej 10 tysięcy). Jest to trudne, bo API wielu serwisów się zmienia lub blokuje narzędzia scrapujące. Rok temu paczka snsrape świetnie pobierała tweety, ale od paru miesięcy nie jest to możliwe. Głównym wyzwaniem tego projektu jest więc zgromadzenie dobrej i dużej bazy danych. Narzędzia do web-scrapingu (np. Selenium, BeautifulSoup) mogą się okazać dobre. Istnieją też gotowe bazy danych tweedów (np. na Kaggle) ale są one już dość mocno wyeksploatowane.

Tematyka bazy danych może być naprawdę różnorodna i obejmować różne dyscypliny życia. Przykłady pytań.

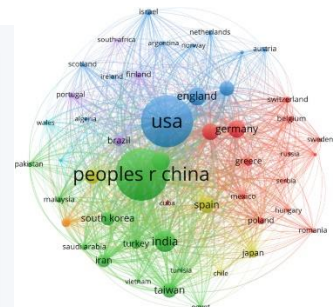
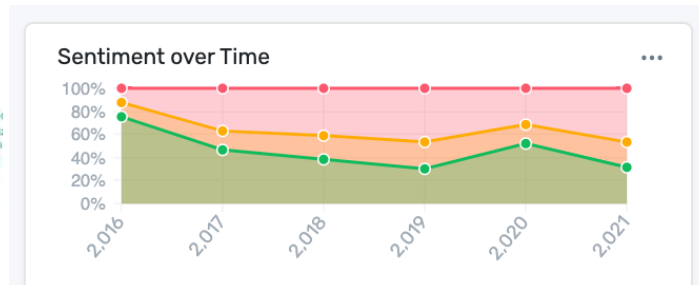
- **Rewolucja AI i emocje.** Jak zmienia się nastawienie ludzi do sztucznej inteligencji na przestrzeni ostatnich miesięcy? #chatgpt #ai #genai

- **Sondaże wyborcze.** Która partia jest bardziej lubiana przed nadchodzącymi wyborami do Parlamentu Europejskiego? Czy natężenie opinii pokrywa się z sondażami wyborczymi? #KO #PIS #KoalicjaObywatelska #PrawoISprawiedlowsco #Tusk #Kaczynski
- **Musk vs Bitcoin, Musk vs Tesla.** Czy opinie nt Elona Muska wpływają na akcje Tesli lub cenę Bitcoina? Sprawdź czy istnieje korelacja w czasie. #Musk #ElonMusk #Tesla #Bitcoin
- **Witcher? Rings of Power? Game of Thrones?** Wybierz popularny serial, najlepiej budzący różne emocje i sprawdź jakie emocje wywołuje wśród użytkowników mediów społecznościowych. Można pomierzyć emocje przed i po konkretnym sezonie, lub konkretnym odcinku i sprawdzić jak zmienia się nastawienie.

Jak widać tematy mogą być polityczne, z showbiznesu, rozrywki czy finansów. Wybierz taki, który Cię szczególnie interesuje. Opinie najlepiej zebrać z różnych okresów w czasie. Musisz przemyśleć jak je rozbić? Z różnych dni/tygodni/miesięcy? A może wystarczy rozbić na dwa okresy: przed i po jakimś wydarzeniu.

Zebrane posty można analizować pod różnymi kątami:

- Jakiek słowa w nich występują? Bag of words, Word cloud, Word frequency.
- Jaki wydźwięk mają? Vader/Text2Emotion itp. Czy wydźwięk się zmienia w czasie? (Wykres liniowy).
- Analiza tematyki opinii i klasteryzacja.



#### 4) PREDYKCJA LUB GENERACJA DLA SZEREGÓW CZASOWYCH (TEKST, MUZYKA, MOWA, GIEŁDA, ITP.)

Jak tytuł wskazuje, chcemy wybrać jakiś typ danych:

- Tekst (z jakiego źródła? Książki? Źródła internetowe?)
- Muzyka (MIDI? MP3? Jaki format? Jakiek źródło?)
- Dźwięk (Mowa? Odgłosy zwierząt?)
- Dane finansowe (Ceny akcji?)

Następnie na podstawie tych danych możemy wyuczyć jakiś model przewidywać lub generować kolejne słowa/dźwięki/nuty/itp., albo klasyfikować dany obiekt (pisarz dla książki, genre dla kawałka muzycznego, gatunek ptaka dla odgłosu, ryzyko inwestycyjne dla akcji giełdy, itp.).

W tym projekcie warto przetestować poznane modele deep learning, czyli:

- Sieci rekurencyjne (zwykłe).
- Sieci LSTM.
- Transformer.

Szczególnie interesujące (zwłaszcza w kontekście tekstu) może być wykorzystanie gotowych modeli (np. pobranych z huggingface) i dotrenowanie ich na naszej bazie danych. Taki fine-tuning może być jednak obciążający czasowo, ale są różne techniki, które przyspieszają trenowanie, np. LoRA (Low Rank Adaptation). W projekcie warto uwzględnić jeden taki eksperyment z fine-tuningiem.

W tym projekcie kuszące są różne zastosowania aplikacyjne. Przykład: trenuję chatbota, którego umieszczam w grze RPG jako postać NPC. Gracz może rozmawiać z taką postacią na różne tematy, ale ważne żeby chatbot zachowywał się jakby był faktycznie postacią w RPGu. Chatbot na stronę UG, który pomagałby użytkownikom strony znaleźć przydatne informacje też byłby ciekawym pomysłem.

Jeśli chodzi o modele tekstowe to dość popularne są:

- Llama <https://huggingface.co/meta-llama/Meta-Llama-3-8B>
- Mistral [https://huggingface.co/docs/transformers/model\\_doc/mistral](https://huggingface.co/docs/transformers/model_doc/mistral)
- Polski model: Bielik <https://huggingface.co/speakleash/Bielik-7B-Instruct-v0.1>

## REALIZACJA PROJEKTU

Kilka zasad realizacji projektu:

- Projekt można realizować w dowolnym języku, Python nie jest wymagany.
- Projekt realizujemy samodzielnie. W szczególnych przypadkach (np. spory stopień skomplikowania) można za zgodą prowadzącego realizować w parach.
- Projekt powinien być unikalny dla każdej osoby (zespołu) w grupie. Prowadzący założy konwersację dla każdej grupy, gdzie należy rezerwować swój temat (kto pierwszy ten lepszy).
- Jeśli prowadzący zajęcia sobie tego zażyczy, należy projekty przechowywać na repozytorium Github/Gitlab/Bitbucket i dodać prowadzącego do repozytorium.
- Czas na wykonanie projektu: do końca semestru, ostatnich zajęć.
- Projekty będą rozliczane podczas zajęć. Możliwe jest indywidualne prezentowanie rozwiązania prowadzącemu zajęcia, lub prezentacja na rzutniku przed grupą. Zaleca się, by przynajmniej parę osób zaprezentowało swój projekt publicznie.
- Projekt może być kontynuacją projektu 1, jeśli rozszerzymy go o wymagane techniki poznane w drugiej części semestru i uwzględnione w tym pliku.
- Projekt może przyjąć formę wyczerpującego raportu badawczego, do którego dołączone będą skrypty pythonowe. Można też zastosować podejście aplikacyjne, tzn. zamiast prostych skryptów stworzyć aplikację (lub jej szkielet) i wówczas raport badawczy może być prostszy i krótszy, i przyjąć formę dokumentacji.

## OCENA PROJEKTU

Niestety w przypadku projektów o różnej tematyce ciężko wyznaczyć jednoznaczną miarę oceniania. Prowadzący zajęcia postarają się wystawić sprawiedliwe punkty, biorąc pod uwagę następujące kryteria:

- Czy wkład studenta (czas, energia, własny kod programistyczny) w projekt był duży czy mały?
  - Tutaj warto, żeby student wskazał co jest zrealizowane samodzielnie, co skopiowane z samouczka, a co wygenerowane przez AI. Należy też uwzględnić wszystkie źródła, z których korzystaliśmy.
- Czy projekt jest oryginalny/nowatorski, czy projekt jest raczej dobrze zbadany/odtwórczy?
- Czy projekt jest dobrze zrealizowany (zawiera wyczerpujące porównanie algorytmów, algorytmy porównane są przejrzystymi miarami)?
- Czy projekt sięga po stare oklepane schematy, czy raczej student starał się korzystać z najnowocześniejszych technik, algorytmów udostępnianych w artykułach naukowych, blogach naukowych itp.?
- Czy student był w stanie dobrze zaprezentować projekt?

## INSPIRACJE DO TEMATÓW

TO DO... (DOPISZĘ PÓŹNIEJ, CHOĆ PEWNIE BĘDĄ TUTAJ NUDNE RZECZY 😊)