

ZROZUMIEĆ AI W PRAKTYCE

Paweł Rychlikowski, Radosław Błasiak
Uniwersytet Wrocławski, Liga Niezwykłych Umysłów
prych@cs.uni.wroc.pl, radoslaw.blasiak@t-lem.com, t-lem.com

Abstract. *This paper presents Python-based AI courses, developed for the League of Extraordinary Minds platform. Covering topics from classical AI and ML to NLP and game-based agents, the curriculum emphasizes conceptual clarity and hands-on learning. Through interactive tasks, students explore key ideas such as search algorithms, heuristics, neural networks, and language models. The approach balances depth with accessibility, aiming to make complex AI topics understandable without oversimplification.*

1. Wprowadzenie

Znajomość metod sztucznej inteligencji wydaje się obecnie bardzo istotna dla każdego członka społeczeństwa. Sądzymy, że bardzo ważne w edukacji jest tworzenie materiałów o jak najwyższej jakości, które umożliwiłyby zapoznanie się z tą problematyką jak największej liczbie zainteresowanych uczniów. Z jednej strony obserwujemy bardzo dynamiczny rozwój technologii AI, z drugiej wydaje się on bardzo skoncentrowany na pewnych metodach AI (związanych z dużymi modelami językowymi). Oczywiście modele językowe z nami pozostaną, ale trudno odpowiedzialnie prognozować, czy za 10 lat będą ciągle awangardą AI. Dlatego optujemy za szerokim rozumieniem pojęcia sztucznej inteligencji, zgodnej z wizją autorów podręcznika *Artificial Intelligence Modern Approach* [1] (4th Edition, 2020). Chcielibyśmy dostarczyć uczniom możliwie kompleksowego warsztatu pojęciowego, który pozwoli odnajdywać się zarówno w klasycznych metodach związanych z przeszukiwaniem czy symulacjami, uczeniem maszynowym z sieciami neuronowymi i bez nich, programowaniem agentów samodzielnie się uczących, czy wreszcie z zadaniami przetwarzania języka naturalnego i (oczywiście) modelami językowymi.

Zależy nam, aby oddać w ręce nauczycieli i uczniów nowoczesny materiał dydaktyczny, który w usystematyzowany, przystępny i przede wszystkim praktyczny sposób będzie prowadził użytkownika przez zagadnienia związane z tworzeniem technologii AI. Wykorzystanie w tym celu profesjonalnej, webowej platformy e-learningowej The League of Extraordinary Minds w istotny sposób uprości proces dydaktyczny, zwalniając nauczycieli z konieczności instalacji i obsługi skomplikowanej infrastruktury sprzętowo-

informatycznej w szkole, a organy prowadzące z jej finansowania. Pozwoli to przeznaczyć czas nauczycieli i zasoby finansowe na efektywne podnoszenie kompetencji uczniów i samych nauczycieli zamiast na zmaganie się z otoczeniem. Uważamy, iż brak zebranego w jednym miejscu, łatwego w użyciu i atrakcyjnego w formie kompendium wiedzy połączonego z praktycznym podejściem do procesu dydaktycznego jest istotną barierą w rozwoju kompetencji uczniów i nauczycieli w zakresie sztucznej inteligencji. Naszym celem jest dostarczenie takiego rozwiązania.

Sądzymy, że nawet o relatywnie trudnych zagadnieniach związanych ze sztuczną inteligencją da się mówić unikając zarówno zbyt daleko idących uproszczeń, jak i formalizacji wychodzących poza możliwości przeciętnego licealisty. Planujemy wprowadzenie na platformie Liga Niezwykłych Umysłów czterech kursów, z których wszystkie korzystają z Pythona. Wybór tego języka jest podyktowany dwoma czynnikami: zgodnie z TIOBE Index (<https://www.tiobe.com/tiobe-index/>) jest to obecnie najbardziej popularny język programowania, a z drugiej strony, ze względu na łatwość kodowania oraz wielką liczbę bibliotek używanych w sztucznej inteligencji (na przykład Scikit-learn <https://scikit-learn.org/>, albo PyTorch).

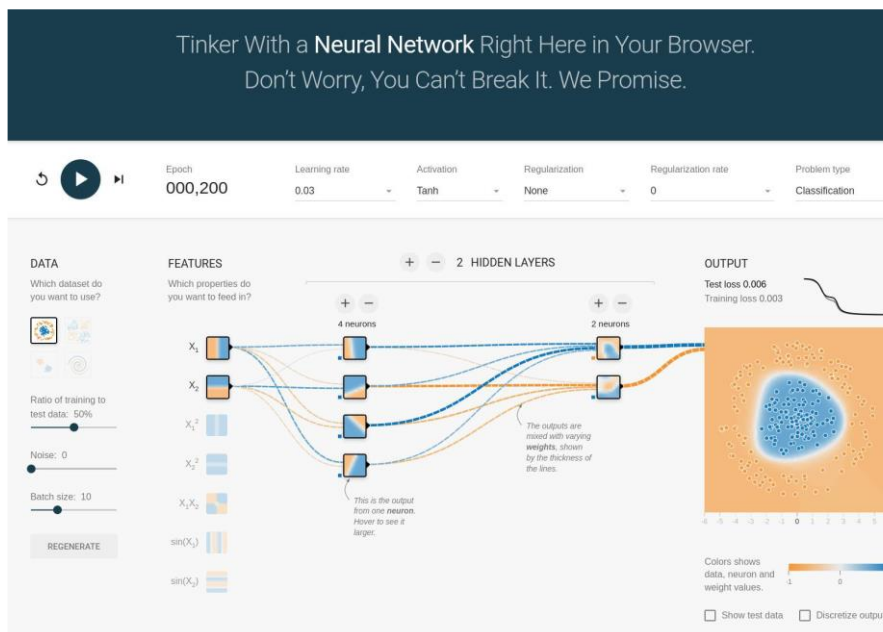
Materiał podzielony zostanie na cztery kursy, z których jeden zostanie udostępniony w lipcu 2025 r. (klasyka sztucznej inteligencji), a kolejne są w zaawansowanej fazie projektowej. Niniejszy artykuł omówi wszystkie kursy, kończąc przedstawienie na kursie przygotowanym do udostępnienia.

2. Kurs 1: Uczenie maszynowe

Ten kurs będzie pokazywał równolegle dwa aspekty uczenia maszynowego: pierwszy dotyczy dogłębnego zrozumienia mechanizmów, jakie stoją za tym pojęciem, drugi natomiast będzie pokazywał zastosowanie metod uczenia maszynowego w możliwie najbardziej życiowych problemach.

Główny nacisk będzie położony na algorytmy klasyfikacji, czyli przypisywania konkretnym obiektom pasujących do nich klas. Podstawowa metoda dydaktyczna będzie nawiązywała do narzędzia pokazującego działanie sieci neuronowych na przykładzie klasyfikacji binarnej (czyli przypisywania do jednej z dwóch grup) punktów na płaszczyźnie. Świetnie ilustruje użyteczność tego pojęcia strona Tensorflow Playground (Rysunek 1), która jednak ma dwie istotne wady:

1. Nie jest pełnoprawnym materiałem e-learningowym, nie zawiera żadnych treści, ani zadań, a ponadto wszystkie zaimplementowane mechanizmy są prezentowane od razu, a nie odsłaniane w momencie, gdy uczeń opanuje wcześniejsze pojęcia.
2. Przedstawia jedynie sieci neuronowe typu MLP, pomijając metody związane z drzewami decyzyjnymi, bardziej zaawansowane sieci neuronowe, maszyny wektorów nośnych, metody zespołowe i wiele innych, relatywnie łatwych do przedstawiania algorytmów.



Rysunek 1. Przygotowana przez firmę Google demonstracja biblioteki TensorFlow. Danymi są zbiory kolorowych punktów, sieć neuronowa wyznacza kolorowanie płaszczyzny na podstawie zdefiniowanych punktów

Nasze doświadczenia dydaktyczne pokazują, że myślenie o "kolorowych punktach" i traktowanie klasyfikatora jako metody „kolorowania płaszczyzny” jest bardzo użyteczne w budowaniu intuicji, daje też świetną kontrolę działania samodzielnie implementowanych programów i umożliwia łatwe usuwanie błędów. Łatwo za pomocą tej metafory tłumaczyć, czym jest zbiór uczący i dlaczego nie wystarczy na nim testować algorytmów, czym jest regularyzacja, przeuczenie, jak możliwe jest uogólnianie i wiele innych zagadnień.

Kurs będzie starał się wyważyć ilość czasu spędzanego na sztucznych danych (głównie w klasyfikacji punktów na płaszczyźnie), jak i na prawdziwych danych (oczywiście wielowymiarowych), możliwie najbardziej interesujących dla uczestników. Niektóre zadania będą wymagały kodowania od podstaw prostych metod, ale będziemy też korzystać z biblioteki sklearn, korzystając z gotowych rozwiązań. Dane syntetyczne uczeń będzie mógł tworzyć za pomocą prostego edytora danych 2D -- i niektóre zadania będą wymagały stworzenia danych, które są szczególnie łatwe, bądź szczególnie trudne dla danego algorytmu.

Szczegółowa lista zagadnień:

- Klasyfikatory liniowe: algorytm perceptronu, regresja liniowa/logistyczna jako proces minimalizacji kosztu, maszyny wektorów nośnych.
- Uczenie się funkcji jednowymiarowej, będącej kombinacją liniową różnych utworzonych sztucznie cech (x , x^2 , ..., $\sin(x)$, $\log(x)$, ...).
- Sieci neuronowe typu MLP. W tym kursie budujemy jedynie wstępne intuicje, bazujące na prostych algorytmach regresji z poprzedniego punktu, stąd sieci wielowarstwowe będą miały losowane wagi pośrednie, a jedynie ostatnia warstwa będzie uczona. Uczeń będzie mógł porównać tak powstałą sieć z wynikiem działania algorytmu propagacji wstecznej.
- Drzewa decyzyjne, ich indukcja, metody zespołowe, czyli jak wiele drzew jest w stanie pokonać jedno drzewo: bagging, lasy losowe, AdaBoost.
- Praktyka pracy z prawdziwymi danymi: dane w różnej skali, niebalansowane dane uczące, brakujące wartości, itp.

3. Kurs 2: Praca z językiem i modele językowe

Znaczna część współczesnej sztucznej inteligencji łączy się z modelami językowymi (LLM). W tym kursie chcemy wprowadzić ucznia w świat przetwarzania języka naturalnego, dając mu szansę na zrozumienie podstawowych mechanizmów LLM-ów, jak również umieścić tę wiedzę w kontekście tradycyjnych metod NLP.

Głównymi motywami kursu będą:

- zadania klasyfikacji tekstu (na przykład rozpoznawanie autorstwa tekstu, segregacja newsów),
- klasyfikacja fragmentów tekstów (znajdowanie danych wrażliwych czy też znajdowanie opisów bohatera/bohaterki jakiejś lektury),
- generacja tekstów (w tym tekstów spełniających określone warunki, na przykład tekstów rymowanych, zawierających tylko wyrazy na określoną literę itp.).

Wprowadzone będą następujące narzędzia:

- naiwny klasyfikator bayesowski i tekst jako 'worek słów',
- n-gramowy model języka,
- word2vec -- wektorowe reprezentacje słów, dlaczego król-mężczyzna+kobieta = królowa,
- proste sieci neuronowe wykorzystywane w pracy nad tekstem,
- mechanizm uwagi, czyli średnia ważona wektorów reprezentujących słowa,
- generacja tekstów za pomocą modeli językowych.

Kurs nie będzie korzystał z żadnego zewnętrznego modelu językowego, zamiast tego będzie używał niewielkich modeli typu GPT, wytrenowanych specjalnie na potrzeby

kursu -- na przykład na powieściach fantastycznych czy kryminalnych z Wolnych Lektur oraz odpowiednio dostrajanych modeli ogólnych (o otwartych źródłach), o wielkości GPT-2. Te modele będą dostrajane do dialogów, ewentualnie do podawania informacji z określonej dziedziny.

Uczeń będzie mógł w różnych zadaniach poznać, jak "od podszewki" działa generacja, czym jest token, jak kontrolować losowość generacji, jak gwarantować, żeby wygenerowane teksty miały oczekiwane właściwości. Cały kurs będzie bazował na podzbiore informacji ze świetnego i aktualnego podręcznika *Speech and Language Processing* (3rd edition) [2].

4. Kurs 3: Sztuczna inteligencja w grach planszowych

W sztucznej inteligencji wiele przełomowych momentów związane było z grami. Można tu wymienić pierwszy system uczący się z gier (Warcaby)[3], zwycięstwo z mistrzem świata w królewskiej grze w szachy (program Deep Blue), wykorzystanie głębokich sieci neuronowych i algorytmu Monte Carlo Tree Search w Go (Mastering the game of Go with deep neural networks and tree search)[4], a z nowszych prac algorytm DeepNash[5], działający w grach o bardzo dużej złożoności i z bardzo ograniczoną wiedzą na temat stanu gry.

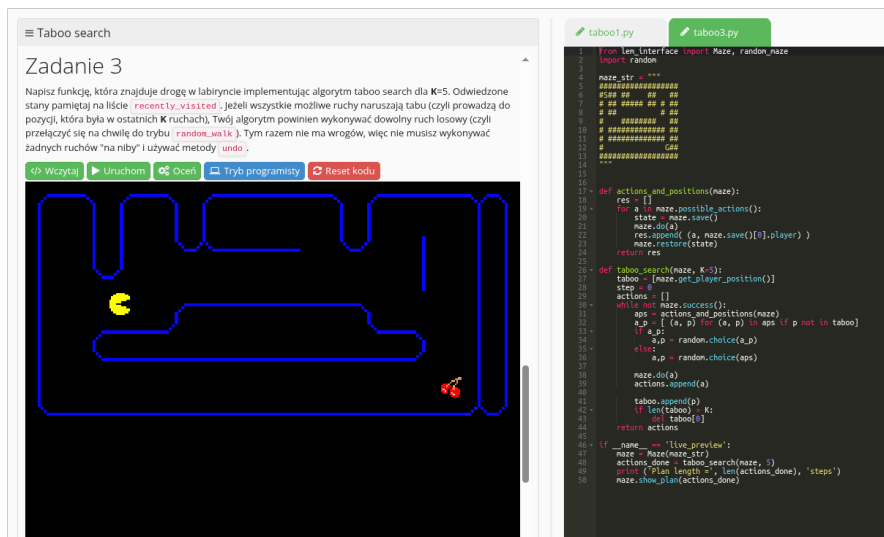
W tym kursie uczeń będzie mógł się zapoznać z szeregiem zagadnień związanych ze sztuczną inteligencją w kontekście gier planszowych. Zagadnienia będą ilustrowane na przykładzie popularnych gier (między innymi warianty kółka i krzyżyka, Connect-4, Animal Chess, Breakthrough). W każdej z gier będzie możliwość testów implementowanego agenta i pewnych ustalonych agentów przygotowanych przez autorów kursu. W niektórych grach będą rozważane dodatkowo warianty z losowością i/lub z niepełną informacją. W dalszej kolejności planujemy stworzyć arenę, w której programy uczniów będą mogły ze sobą współzawodniczyć w specjalnie utworzonych na potrzeby kursu grach planszowych.

Wybrane zagadnienia z kursu:

1. MiniMax i Alpha-Beta pruning
2. Heurystyczna ocena planszy
3. MCTS w grach
4. Uczenie się za pomocą self-plays
5. Regresja liniowa i uczenie współczynników do heurystycznej oceny planszy
6. Podstawy wykorzystania sieci spłotowych (konwolucyjnych) w grach
7. Algorytm DeepNash i praca z „mgłą wojny”

5. Kurs 4: Klasyka sztucznej inteligencji w świecie Pac-Mana

W tym kursie będziemy pracować nad różnymi wariantami świata gry Pac-Man. W kursie znajduje się kilkadziesiąt zadań programistycznych o zróżnicowanym charak-



Rysunek 2. Przykładowy widok podczas pracy z kursem w jednej z początkowych lekcji

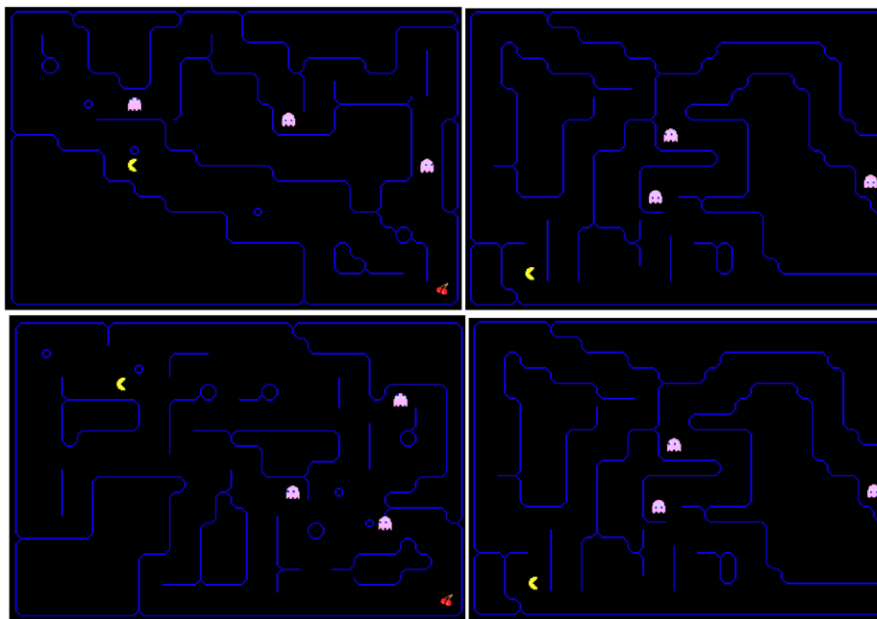
terze. Dodatkowo zakładamy, że wszystkie warianty gry Pac-Man, które pojawią się w kursie będą dostępne również jako mini-gry (a raczej łamigłówek), które uczeń będzie mógł samodzielnie rozwiązywać i „ścigać” się na wyniki z implementowanymi następnie agentami. Dodatkowym efektem będzie zauważenie, że trudność pewnych zadań dla człowieka i komputera nie zawsze jest skorelowana i rzeczy łatwe dla człowieka mogą być istotnie trudniejsze dla programu komputerowego i vice versa.

Kurs zawiera 12 lekcji, mówiących o różnych wariantach konstruowania agentów, sposobach przeszukiwania, o modelowaniu świata deterministycznego i niedeterministycznego, o sposobie wykorzystania heurystyk i wykorzystaniu symulacji. W każdej lekcji jest od 3 do 5 zadań programistycznych, ze specjalnie przygotowanymi testami i procedurami automatycznego sprawdzania. Żeby lepiej utrzymać uwagę ucznia, świat Pacmana ulega przeobrażeniom w trakcie kursu. I tak lekcje 1-4 zakładają pracę z pustym labiryntem, w którym jedynymi obiektami są Pacman, nieruchomi wrogowie i cel do którego dążymy (Rysunek 2). W lekcji piątej pojawiają się ruchomi przeciwnicy, realizujący różne mechaniki. Od lekcji 7 zmienia się cel -- nie jest to już dotarcie do konkretnego punktu, lecz zebranie wszystkich kropek na planszy (tak jak w oryginalnej grze). Od lekcji 9 zaczynają nam towarzyszyć wrogowie poruszający się do pewnego

stopnia losowo (przy czym sukcesywnie wprowadzane są coraz to bardziej wymagające mechaniki ich działania). Ostatnia lekcja wprowadza z kolei ideę multiwersum, czyli jednoczesnego sterowania kilkoma agentami, poruszającymi się w różnych segmentach planszy.

1. Wprowadzenie do świata Pacmana i spacerów losowe: w tej lekcji uczeń może zapoznać się ze światem Pacmana i ze sposobem, w jakim będziemy tworzyć agentów w tym świecie. Podstawowym wprowadzonym pojęciem będą spacerów losowe (czyli będziemy badać agenty, które wykonują losowe ruchy). Będzie mowa o obliczaniu prawdopodobieństwa sukcesu algorytmu bazującego na losowości oraz nieuniknionej zmienności działania takich algorytmów.
2. Jak nie kręcić się w kółko, czyli taboo search: uczeń, który zidentyfikował problemy agenta losowego, będzie mógł zapoznać się z prostą modyfikacją, która pozwala kilkukrotnie poprawić działanie spacerów losowych: mianowicie z algorytmem taboo search, który poprawia właściwości eksploracyjne. Podczas tej lekcji uczeń zdobędzie również podstawowe intuicje dotyczące dostrajania hiperparametrów, regulujących działanie algorytmu.
3. Prosto do celu, czyli przeszukiwanie zachłanne: uczeń zapozna się z prostymi sposobami ukierunkowania przeszukiwań ze względu na położenie celu. Pozna, że czyista zachłanność często prowadzi do minimum lokalnego i pozna sposoby radzenia sobie z tym problemem. Ta lekcja przybliży istotny w SI dylemat eksploracja-eksploatacja (czy badać świat, wykonując nieoptymalne ruchy, czy też jak najlepiej wykorzystywać posiadaną wiedzę. Omawiane algorytmy to między innymi: hill climbing, hill climbing with random restarts, symulowane wyżarzanie oraz ich warianty i kombinacje.
4. Zróbmy to porządnie, czyli przeszukiwanie bez losowości: w tej lekcji przedstawione zostaną klasyczne algorytmy sztucznej inteligencji, związane z wyszukiwaniem drogi w grafie. W szczególności omówimy przeszukiwanie w głąb, przeszukiwanie wszerz, iteracyjne pogłębianie. Przedstawione również zostanie znaczenie losowości w deterministycznych algorytmach.
5. Pacman uciekający przed duszkami, czyli o tym, że BFS to coś więcej niż przeszukiwanie labiryntów: podczas tej lekcji uczeń ma okazję przekonać się, że BFS działa nie tylko w labiryntach, ale również w bardziej abstrakcyjnych przestrzeniach, opisujących różne możliwe konfiguracje świata. Ze względu na obecność przeciwników, stanem nie jest już jedynie pozycja Pacmana -- konieczne staje się włączenie do niego informacji o duszkach (ich położeniu, a w pewnych wariantach również kierunku ruchu). Zrozumienie abstrakcji stanu, kluczowe dla zrozumienia bardzo wielu algorytmów AI, od tego momentu będzie istotne w praktycznie wszystkich lekcjach. Pewną formą „nagrody” dla ucznia będzie możliwość obserwacji, jak algorytmy działające w zwykłych, statycznych labiryntach przenoszą się z minimalnymi zmianami do przestrzeni stanów -- ale najkrótsza droga w tej przestrzeni nie jest

najkrótszą drogą w labiryncie. Można obserwować, jak agent wabi swoich przeciwników w pułapki, próbuje ich eliminować doprowadzając do sytuacji, w której więcej wrogów jest w jednym miejscu itd. Dzięki możliwości testowania algorytmów w losowanych labiryntach uczeń ma możliwość lepszego przekonania się o uniwersalności zaimplementowanych algorytmów.



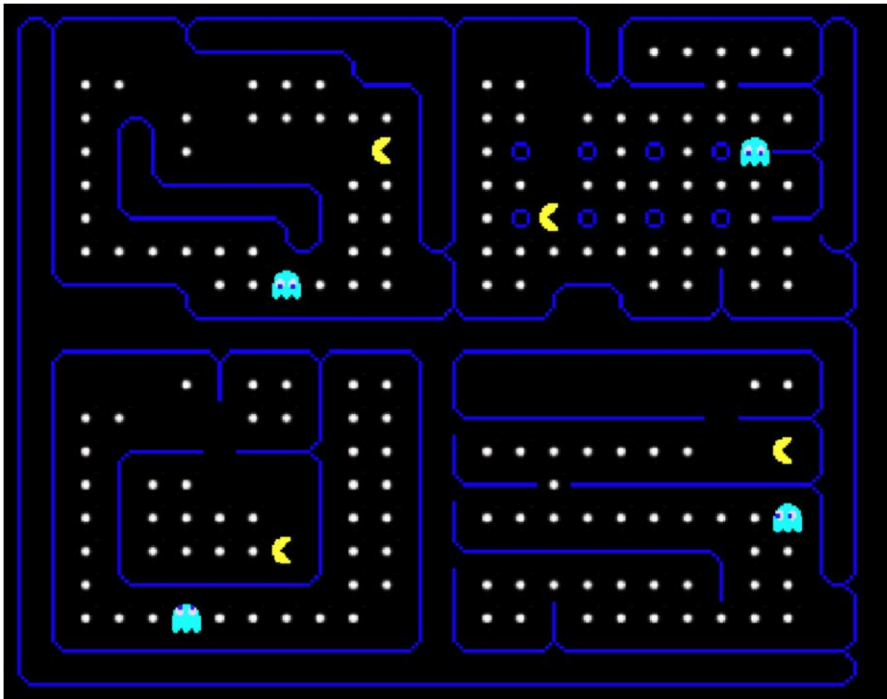
Rysunek 3. Agent w trakcie rozwiązywania czterech wylosowanych labiryntów

Na rysunku 3 przedstawione są przykładowe wylosowane labirynty.

6. Tworzenie nowych światów, czyli wykorzystanie BFS w projektowaniu labiryntów: w tej lekcji zmieniamy perspektywę -- agent jest dany, a uczeń ma napisać algorytm konstruujący świat, w którym dany agent: a) ma moc znaleźć rozwiązanie, b) to rozwiązanie ma być możliwie jak najbardziej wymagające (oczywiście przy danej wielkości świata i innych jego ograniczeniach).
7. O zjadaniu kropek, czyli o uniwersalności i granicach algorytmu BFS: głównym celem tej lekcji jest przedstawienie kombinatorycznej eksplozji stanów. Jeżeli bowiem stanem jest zbiór kropek, to liczba różnych stanów, które analizuje algorytm jest bardzo duża i efektywnie da się rozwiązywać jedynie najprostsze układy. W tej lekcji będzie mowa również o heurystycznych sposobach radzenia sobie z tą sytuacją.

8. Algorytm A*, czyli jak wiedza o zadaniu pomaga w walce z wykładniczą eksplozją stanów: podczas tej lekcji uczeń ma poznać klasyczny algorytm AI mianowicie A*, który jest modyfikacją BFS-a, umożliwiającą wykorzystanie wiedzy o problemie. Algorytm wykorzystuje heurystyczne szacowanie odległości bieżącego stanu od stanu końcowego i podczas lekcji będzie możliwość analizy kilku sposobów wykonywania takich szacowań
9. Błądzące duszki, czyli jak proste odruchy mogą dać inteligentne zachowanie: w tej lekcji pojawia się losowość wrogów, co oznacza, że nie istnieje już coś takiego jak liniowy plan (sekwencja akcji) prowadząca do sukcesu -- agent musi podejmować decyzje w każdej turze, zapoznawszy się z aktualną sytuacją w labiryncie. W tej lekcji uczeń będzie miał okazję, zbadać, jak zachowują się agenty, które decyzję podejmują bez pogłębionej, strategicznej analizy sytuacji, realizując różne warianty „polityki odruchów” (reflex policy).
10. Pacman idzie do kasyna, czyli ocenianie stanów za pomocą symulacji: w tej lekcji wprowadzamy ocenę sytuacji za pomocą wielokrotnego symulowania potencjalnych zmian świata i uśredniania ich wyników. Uczniowie mają okazję zapoznać się z tą ideą najpierw na statycznych labiryntach, a następnie w labiryntach z bardziej wymagającymi przeciwnikami, których zachowanie z jednej strony jest niedeterministyczne, ale z drugiej bardziej "inteligentne" niż w poprzedniej lekcji.
11. Co łączy Pacmana, Koreę i Międzynarodową Olimpiadę Matematyczną, czyli Monte Carlo Tree Search: problemem prostego Monte Carlo jest nieefektywne gospodarowanie budżetem obliczeniowym przeznaczonym na symulację. W tej lekcji pokażemy algorytm MCTS, który rozwiązuje ten problem, tworząc efektywne drzewo przeszukiwań. Ponieważ algorytm jest dość złożony, uczeń będzie do niego dochodził stopniowo, rozwiązując kilka syntetycznych zadań związanych z pracą z drzewami (i ich rozrostem), wyborem odpowiednich węzłów do rozwinięcia i tworzeniem statystyk w węzłach. Pełen algorytm będzie uruchamiany na nowym wariantcie świata, w którym wrogowie zachowują się na różne sposoby.
12. Ostatnia lekcja: w stronę multiwersum: w ostatniej lekcji pozwolimy uczniowi na znaczną swobodę doboru algorytmów do zadania. Zmieni się jednak świat -- podstawową nową mechaniką będzie obecność kilku Pacmanów, z których każdy operuje w innym fragmencie labiryntu, ale są sterowane wspólnymi poleceniami.

Przykładowy stan działania agenta zbierającego równocześnie kropki w czterech światach pokazuje rysunek 4:



Rysunek 4. Pacmanowe multiwersum. Agent wydaje jedno polecenie (na przykład: idź w lewo) do wszystkich czterech Pacmanów jednocześnie. Przykład złożonego świata z ostatniej lekcji.

Literatura

1. Russel S., Artificial Intelligence: A Modern Approach, Pearson, 2020
2. Jurafsky D., Martin J., Speech and Language Processing (3rd edition), 2025
3. Samuel A. L., Some Studies in Machine Learning Using the Game of Checkers, IBM, 1959
4. Praca zbiorowa, "Mastering the game of Go with deep neural networks and tree search", Nature, 2016, <https://www.nature.com/articles/nature16961>
5. Praca zbiorowa, "Mastering the game of Stratego with model-free multiagent reinforcement learning", Science, 2022, <https://www.science.org/doi/10.1126/science.add4679>