



**Wydział Inżynierii Mechanicznej i
Robotyki**

Katedra Robotyki i Mechatroniki



Podstawy Sztucznej Inteligencji I Uczenia Głębokiego *Inżynieria Mechatroniczna*

Instrukcja 12:

Zaawansowane modele AI

Nauczysz się: Jak działają wybrane narzędzia z obszaru sztucznej inteligencji. W zależności od Twoich preferencji nauczysz się więcej na temat używania głębokich modeli językowych, modeli GenAI tworzących obrazy lub głębokich konwolucyjnych sieci neuronowych do klasyfikowania danych inżynierskich.

Dodatkowe materiały:

- *Finalne wykłady przedmiotowe o numerach od 10 do 14*

Instrukcja przetłumaczona przez ChatGPT z angielskiego oryginału – materiałów z przedmiotu Basics of Artificial Intelligence and Deep Learning

Koordynator przedmiotu:

Krzysztof Holak, holak@agh.edu.pl

Autor instrukcji:

Ziemowit Dworakowski, zdw@agh.edu.pl

Adam Machynia, machynia@agh.edu.pl

Wprowadzenie

To laboratorium przebiega w inny sposób niż wszystkie pozostałe zajęcia na tym kursie.

Po pierwsze, zostało ono zaplanowane na dwa spotkania – rozpoczniesz pracę podczas jednego (12. laboratorium), a zakończysz ją podczas kolejnego (13. laboratorium). Z tego powodu instrukcja zawiera zarówno zadania do wykonania podczas zajęć, jak i elementy pracy własnej w domu. Idea polega na tym, aby najpierw poznać podstawowe zagadnienia i omówić z prowadzącym wstępne podejście do problemu, następnie przeprowadzić samodzielne eksperymenty w domu, a na końcu – przedstawić i omówić wyniki z prowadzącym oraz z kolegami i koleżankami z grupy podczas ostatniego spotkania.

Po drugie, instrukcja zawiera 8 oddzielnych tematów do wyboru. Twoim zadaniem jest realizacja tylko jednego z nich. Zapoznaj się z wszystkimi, ale tym razem nie wykonuj żadnej pracy wcześniej – tematy zostaną przydzielone podczas 12. laboratorium przez prowadzącego (prawdopodobnie w oparciu o dotychczasowe wyniki – osoby z wyższymi ocenami z testów i laboratoriów będą mogły wybrać temat jako pierwsze). Zwróć uwagę, że niektóre tematy będą się rozwijały nieco inaczej dla osób korzystających z darmowych oraz płatnych rozwiązań generatywnej AI. Wybór jednej z nich zależy wyłącznie od Twoich preferencji i nie wpływa na ocenę. Modele różnią się często istotnie – niektóre zadania mogą nie być wykonalne w darmowych wersjach. Jeśli posiadasz subskrypcję, warto ją wykorzystać i nie ograniczać się niepotrzebnie.

Po trzecie, ponieważ realizacja zadań laboratoryjnych będzie czasem wymagać wielu eksperymentów – w niektórych tematach możliwa jest również praca w parach. W takim przypadku wymagane będzie wykonanie nieco większej liczby testów lub ocen (np. więcej prób lub analiz). Szczegółowe informacje znajdują się w opisie każdego z tematów – zwróć uwagę, że niektóre tematy nie dopuszczają pracy zespołowej. Ponadto część zadań umożliwia pracę z różnymi zestawami danych – dlatego mogą być realizowane przez więcej niż jedną osobę (liczba dostępnych „slotów” jest podana przy temacie).

Zasada oceniania pozostaje niezmienną – im więcej wykonanych zadań, tym wyższa ocena końcowa. Tym razem jednak nie przewidujemy dokończenia zadań w domu po ostatnim laboratorium – ocena uzyskana podczas 13. spotkania zostanie uznana za ostateczną (o ile będzie pozytywna).

Tematy do wyboru:

1. Ogólna interakcja z ChatGPT (tylko jedna osoba, wymaga subskrypcji)

Nauczysz się, jak tworzyć skuteczne prompty, testować różne funkcjonalności ChatGPT oraz poznasz bieżące ograniczenia jego kompetencji. Temat będzie wymagał zapoznania się z dodatkowymi materiałami związanymi z wybranym zagadnieniem – a wyższe oceny będą wymagały refleksji nad działaniem dużych modeli językowych na meta poziomie.

2. Interakcja z modelami generującymi obrazy (tylko jedna osoba)

Wybierzesz i przetestujesz wybrany model generatywny (genAI) pod kątem realizacji różnych zadań, starając się zachować spójność wyników i jak najdokładniej odwzorować zamierzony efekt. Temat ten szczególnie zyskuje, jeśli posiadasz subskrypcję w jednym z płatnych narzędzi (np. Midjourney lub Dalle3) – umożliwi to bowiem znacznie mniej ograniczoną liczbę prób.

3. Klasyfikacja danych z użyciem wyłącznie LLM (1-2 osoby)

Ponownie zbudujesz płytkie i głębokie klasyfikatory danych w oparciu o zestawy danych używane podczas kursu. Tym razem jednak wszystkie operacje wykonywać będzie za Ciebie „LLM-owy sługa”. Twoim zadaniem będzie definiowanie zadań oraz ocena poprawności generowanych wyników.

4. Pozyskiwanie informacji naukowych z wykorzystaniem wyspecjalizowanych LLM (2x jedna osoba)

To temat specjalistyczny, w którym nauczysz się, jak korzystać z dużych modeli językowych do pozyskiwania informacji naukowych, jak badać stan wiedzy w danym obszarze oraz jak weryfikować, czy uzyskane dane są wiarygodne i bezpieczne do użycia. Temat będzie wymagał zapoznania się z dodatkowymi materiałami merytorycznymi.

5. Demonstrator „podstawowej sieci GAN” (1-2 osoby)

Wszystko zaczęło się od podstawowych sieci generatywno-przeciwstawnych (GAN) – a przynajmniej wszystko, jeśli chodzi o rozwój współczesnych modeli generatywnych. W tym temacie spróbujesz zbudować własny, podstawowy model GAN do generowania obrazów według własnego wyboru. To dobre wprowadzenie do praktycznego uczenia się o modelach generatywnych, ale nie należy spodziewać się efektownych wyników jak te, które można znaleźć w Internecie – to w końcu stary i dość prosty model, który prawdopodobnie będzie wymagał wielu godzin na osiągnięcie sensownych rezultatów.

6. Projektowanie gry z użyciem LLM (2 x 1-2 osoby)

Stworzysz własną grę w grafice ASCII z pomocą LLM i wybranego języka programowania (Matlab, Python, C++ itd.). Przetestujesz zdolność modelu do utrzymywania kontekstu oraz tworzenia funkcji zgodnych z uprzednio zdefiniowaną strukturą kodu.

7. Interpretacja sygnałów rzeczywistych w dziedzinie czasu (1-2 osoby)

Pamiętasz bazę danych z symulowanymi sygnałami, z której korzystaliśmy podczas 5. instrukcji? W tym temacie wykonasz podobne zadanie, ale na bazie sygnałów rzeczywistych, stosując podejście oparte na głębokim uczeniu. Zaprojektujesz głęboką sieć konwolucyjną do rozwiązania postawionego zadania i porównasz jej działanie z płytkim klasyfikatorem bazującym na cechach. Temat ten będzie wymagał zapoznania się z dodatkowymi materiałami dotyczącymi kontekstu zastosowania – w szczególności z obszaru monitorowania stanu technicznego maszyn.

8. Praktyczny klasyfikator obrazów oparty na danych własnych (2 x 1-2 osoby)

Temat ten wymaga niemal identycznego przygotowania jak to, które realizowaliśmy w laboratorium nr 11 – z tą różnicą, że samodzielnie musisz zebrać dane do realizacji wybranego przez siebie zadania. Choć może to początkowo wydawać się proste, organizacja praktycznego zbierania danych nie jest trywialna – nauczysz się w tym temacie m.in. o źródłach wycieku danych oraz o tym, jak tworzyć dane ogólne i reprezentatywne. Zwróć uwagę, że temat ten wymaga dużo „pracy w terenie” i nie nadaje się dla osób prokrastynujących – nie uda się go zrealizować w kilka godzin dzień przed ostatnim laboratorium.

Temat 1: Interakcje z ChatGPT

Chociaż w temacie wskazano model LLM, możesz – po konsultacji z prowadzącym – użyć innego modelu, jeśli bardzo Ci na tym zależy. Pamiętaj jednak, że wybrany model musi posiadać pamięć interakcji, aby umożliwić budowanie długoterminowego kontekstu pracy.

1. Wybierz jedno z laboratoriów realizowanych podczas kursu. Następnie sformułuj zagadnienia do oceny działania ChatGPT. Powinieneś przygotować:
 - o Zadanie „dokumentacyjne” – np. przygotowanie sprawozdania z laboratorium na podstawie danych, wyników i kodów, które opracowałeś realizując to ćwiczenie „ręcznie”.
 - o Zadanie „programistyczne” – np. stworzenie funkcji realizującej dodatkowe zadanie z instrukcji, którego nie robiłeś w standardowym przebiegu laboratorium.
 - o Zadanie „edukacyjne” – zdobycie wiedzy na temat kontekstu wybranych metod i zagadnień, np. ograniczeń ich stosowania, ryzyk czy potencjalnych obszarów zastosowania. Obejmuje to korzystanie z LLM w celu zdobycia nowej wiedzy w zakresie powiązonym z tematem laboratorium.
 - o Zadanie „kreatywne” – rozszerzenie instrukcji o nowe elementy, np. stworzenie własnego fragmentu instrukcji lub przeprojektowanie jej wybranych części.

Zadania te należy udokumentować i uzyskać akceptację prowadzącego.

2. Zrealizuj zadania wykorzystując „świeży” model (w ChatGPT może to być np. „tymczasowy” czat). Staraj się konstruować jak najlepsze prompty, ale zrezygnuj z podejścia wieloetapowego – wszystkie niezbędne informacje powinny znaleźć się w pojedynczym promptcie. Zapisz użyte prompty oraz uzyskane wyniki w formie raportu. Przeanalizuj krytycznie wyniki i dodaj tę analizę do raportu. Nie używaj meta-podejścia: nie testuj LLM-a i nie każ mu samemu oceniać swojego działania.
3. Teraz wykorzystaj możliwość przechowywania informacji przez model. Opisz mu swoje cele i priorytety, język, jakiego ma używać w dalszej interakcji z Tobą oraz jego rolę w tych interakcjach. Rozwiąż ponownie zadania, tym razem stosując podejście wieloetapowe – zacznij od przekazania kontekstu, sprawdzenia, czy model go zrozumiał, a następnie stopniowo buduj odpowiedź (od ogólnych punktów, przez listę treści, aż do pełnego tekstu lub rozwiązania zadania). Zapisz wyniki i użyte prompty, a następnie porównaj je krytycznie z wynikami z poprzedniego punktu.
4. Przeanalizuj krytycznie zadanie edukacyjne. Znajdź odpowiedzi na postawione pytania w „klasyczny sposób” (np. literatura, podręczniki) i porównaj: (a) jak trafne i poprawne były odpowiedzi modelu, (b) na ile były kompletne względem kontekstu, jaki podałeś. Czy Ty sam wyjaśniłbyś ten materiał inaczej?
5. Wróć do zadania edukacyjnego realizowanego z LLM i zacznij dopytywać o szczegóły i wyjaśnienia. Oceń przydatność modelu w: (a) wskazywaniu dokładnych źródeł podawanych informacji oraz (b) dostarczaniu pomocnych wyjaśnień (doprecyzowywaniu zagadnień zgłaszanych przez Ciebie).
6. Wróć do zadania kreatywnego i wykorzystaj model jako „lustro” pomagające Ci lepiej sformułować własne pomysły. Spróbuj zaproponować celowo dziwne i nie do końca sensowne koncepcje (np. mieszając temat laboratorium z niezwiązanymi dziedzinami lub wprowadzając zawiłe przykłady). Sprawdź, jak dobrze model potrafi wskazać Twoje „błędy” – pod warunkiem, że nie zostanie wcześniej poproszony o ich analizę. Przygotuj raport dokumentujący Twoje obserwacje

Temat 2: Interakcja z modelami generującymi obrazy

Temat ten jest koncepcyjnie łatwiejszy przy użyciu modeli płatnych (wystarczy wpisać prompt i otrzymać obraz), ale może być bardziej satysfakcjonujący w wersji darmowej (wymaga instalacji i uruchomienia własnego modelu). Nie wszystkie zadania da się wykonać przy użyciu każdego modelu. Z tego powodu po zadaniach na „3.0” wszystkie kolejne zadania oznaczone są kolorem zielonym. Aby uzyskać ocenę 4.0, należy rozwiązać przynajmniej jedno dodatkowe zadanie. Aby uzyskać ocenę 5.0 – wystarczy dwa (jeśli zostaną wykonane poprawnie).

1. Wybierz dowolny model do generacji obrazów. Łatwe, ale płatne opcje to np. Midjourney lub Dalle2. Darmowe opcje to np. <https://www.mage.space/> lub modele dostępne np. na <https://huggingface.co/> – ale w tym przypadku wymagane będzie przygotowanie środowiska. Można uruchomić ChatGPT i poprosić o szybki przegląd dostępnych opcji w ramach wybranych ograniczeń. Przed podjęciem decyzji sprawdź stopień dostępności modelu – prawdopodobnie będziesz musiał wygenerować co najmniej 40–50 obrazów, aby zrealizować ten temat.
2. Wybierz tematy dla następujących kategorii: logo (np. dla Twojego przyszłego startupu), zestaw spójnych stylistycznie obrazów ilustrujących różne koncepcje (np. do zilustrowania książki o wybranym hobby lub jako grafiki na stronę internetową) oraz duży plakat na temat związany z obszarem mechatroniki (np. turbiny wiatrowe lub konkretna gałąź robotyki czy elektroniki). Opisz swoje tematy oraz przygotuj zestaw wymagań i wytycznych (załóż, że jesteś osobą zlecającą wykonanie grafiki). **Udokumentuj wymagania i uzyskaj akceptację prowadzącego.**
3. Zrealizuj każde z powyższych zadań w wybranym przez siebie narzędziu do generacji obrazów. Możesz zmieniać narzędzia, jeśli uznasz, że jedno z nich lepiej sprawdzi się w jednym zadaniu, a inne – w kolejnym. Zapisuj prompty oraz przygotuj raport dokumentujący Twój postęp (wraz z obrazami pośrednimi i błędami).
4. Wykorzystaj plakat wygenerowany w zadaniu 2 i spróbuj usunąć z niego konkretne elementy – lub zastąpić je innymi. Sprawdź, jak dobrze model rozumie Twoje intencje i jak skutecznie potrafi zmodyfikować obraz bez wpływania na jego pozostałe części. Przetestuj dostępne strategie rozwiązania tego problemu – np. rozpoczęcie generacji od tego samego ziarna (seed) ze zmodyfikowanym promptem, inpainting na wygenerowanym obrazie, inpainting na ręcznie zmodyfikowanym obrazie. Nie wszystkie strategie będą dostępne we wszystkich modelach. Opisz wyniki i obserwacje.
5. Wykonaj ponownie zadanie z plakatem – tym razem wykorzystując inne modele lub inne konfiguracje dostępne w Twoim środowisku. Przygotuj raport z wynikami i oszacuj, który z modeli lub które z ustawień najlepiej sprawdziły się w realizacji zadania. Należy tu uwzględnić nie tylko „atrakcyjność obrazu”, ale również „czynnik merytoryczny” – czy model rzeczywiście uchwycił wygląd elementów, które miały zostać przedstawione, czy też raczej stworzył „fanfik” na temat mechatroniki?
6. Wybierz opcję LoRA pasującą do wcześniej zdefiniowanego zadania „zestaw obrazów” (uwaga: LoRA nie jest dostępna w każdym środowisku). Sprawdź, czy pozwala ona uzyskać wyższą spójność w zestawie generowanych obrazów. Jeżeli nie możesz użyć LoRA zgodnej z pierwotnym opisem, wybierz najbliższą możliwą i przetestuj wynik. Opisz wyniki i obserwacje w raporcie.
7. Wykorzystaj swoje zdjęcie jako bazę do wygenerowania obrazu związanego z robotyką (np. robot z Twoją twarzą, Ty montujący robota, Ty uciekający przed robotem itp.). Przetestuj różne strategie dostępne w modelu (jeśli je obsługuje): przesyłanie zdjęcia jako inspiracji lub inpainting na Twojej fotografii. Oceń, jak bardzo obraz jest podobny do oryginału. Opisz wyniki w raporcie.

Temat 3: Klasyfikacja danych z użyciem LLM

Wykorzystaj dane używane wcześniej w zadaniach klasyfikacyjnych podczas laboratoriów. Do dyspozycji masz zestawy: „Sztuczne klastry” (A), „Sygnały inżynierskie” (B) oraz „Cechy obrazów” (C). W tym temacie nie będziesz korzystać z naszych instrukcji ani własnych kodów Matlabowych do rozwiązywania problemów klasyfikacyjnych. Zamiast tego wykorzystasz wyłącznie wybrany przez siebie duży model językowy (zalecane: ChatGPT 4.0 lub nowszy). Celem jest załadowanie danych do LLM-a i poprowadzenie procesu ich przetwarzania aż do uzyskania klasyfikatora i etykiet dla zbiorów treningowych oraz walidacyjnych. Uwaga: wnioski z zadania mogą być zupełnie inne w zależności od zastosowanego modelu! Zapisuj całą interakcję z LLM-em — będzie ona omawiana z prowadzącym podczas 13. laboratorium.

1. Weź zbiór danych A, podzielony na zbiory: treningowy, walidacyjny i testowy. Bezpiecznie zapisz zestaw testowy i **nie pokazuj go modelowi**. Przekaż modelowi dane treningowe i walidacyjne, poproś o ich ocenę oraz wygenerowanie wykresów. Zachowaj wygenerowane przez model wykresy.
2. Poproś model o zbudowanie klasyfikatora na podstawie zbioru treningowego. Poproś go o ocenę jakości klasyfikatora oraz optymalizację jego struktury. Gdy uznasz, że działanie modelu jest satysfakcjonujące, przekaz mu zestaw testowy **bez etykiet** – poproś o klasyfikację i pobierz wynik. Następnie ręcznie porównaj etykiety uzyskane od LLM-a z rzeczywistymi etykietami, których wcześniej nie udostępniałeś. Sprawdź ręcznie dokładność klasyfikacji i porównaj z wynikami uzyskanymi podczas wcześniejszych laboratoriów. Przygotuj krótki raport z wynikami i wnioskami.
3. Jeśli chcesz uzyskać wyższą ocenę: dla zbiorów B i C rozpocznij od danych surowych (tylko zbiór treningowy) i poproś model o zaproponowanie algorytmów ekstrakcji cech (możesz ręcznie przygotować kod dzielący dane, by model nie miał dostępu do zbioru testowego). Spróbuj sformułować zadanie tak, by model obliczał te same cechy, które analizowaliśmy podczas laboratoriów. **Nie przekazuj żadnych fragmentów kodu ani instrukcji** – komunikuj się wyłącznie w języku naturalnym.
4. Przetestuj możliwość przeprowadzenia pełnej ekstrakcji cech w trybie online. Czy możliwe jest nakierowanie modelu tak, by samodzielnie przetworzył surowe dane, obliczył macierze cech i zapisał je do dalszego wykorzystania? Jeśli tak – wykonaj to. Jeśli nie – sprawdź, w którym momencie proces się załamuje i wyjaśnij dlaczego.
5. Przeprowadź klasyfikację analogicznie do punktów 1 i 2. Jeśli zrealizowałeś punkt 3 lub 4 – wykorzystaj cechy wygenerowane przez model. W przeciwnym razie – skorzystaj z cech, które zostały wyekstrahowane podczas wcześniejszych laboratoriów. Porównaj wyniki klasyfikacji zbioru testowego z rzeczywistymi etykietami (ground truth).
6. Poproś model o przetestowanie różnych klasyfikatorów (SVM, kNN, modele zespołowe) i wskazanie, który z nich najlepiej pasuje do danego zadania.
7. Przetestuj możliwość wykorzystania uczenia głębokiego do klasyfikacji surowych danych w zbiorach B i C – **bez wcześniejszego wyznaczania cech**.
8. Porównaj wyniki uzyskane trzema podejściami: manualnym (wyniki z laboratoriów), półautomatycznym (LLM tworzy kod, my go uruchamiamy), oraz w pełni automatycznym (LLM tworzy i uruchamia kod, a następnie zwraca gotowe etykiety). Oceń nie tylko dokładność klasyfikacji, ale także **czas poświęcony na realizację każdego z podejść**. Spróbuj oszacować faktyczny nakład pracy.
9. Przygotuj sprawozdanie porównujące dokładność uzyskaną podczas zajęć z dokładnością osiągniętą przy użyciu LLMA. Przedstaw swoje wnioski i obserwacje.

Temat 4: Pozyskiwanie informacji naukowych z wykorzystaniem wyspecjalizowanych LLM

Twoim celem jest przygotowanie przeglądu literatury w oparciu o źródła naukowe. W każdym przypadku należy zapisywać rozmowy prowadzone z modelami LLM.

1. Wybierz temat do przeglądu literatury. Skonsultuj go z prowadzącym i uzyskaj jego akceptację.
2. Doprecyzuj zapytanie wyszukujące i przygotuj wstępne prompty.
3. Przeprowadź wyszukiwanie z wykorzystaniem domyślnego modelu ChatGPT. Pamiętaj, że potrzebujesz odniesień do literatury naukowej. Po uzyskaniu wstępnego przeglądu rozwiń najważniejsze aspekty tematu.
4. Powtórz zadanie 3 z wykorzystaniem Perplexity.ai.
5. Powtórz zadanie 3 z wykorzystaniem Scholar GPT – modelu GPT dostosowanego do pracy naukowej.
6. Powtórz zadanie 3 z wykorzystaniem jednego z następujących narzędzi: <https://scite.ai>, <https://storm.genie.stanford.edu> lub <https://www.undermind.ai>.
7. Porównaj wyniki z zadań 3–6. Zidentyfikuj główne różnice i oceń jakość uzyskanych rezultatów. Wskaż, który model najlepiej odpowiada Twoim potrzebom i uzasadnij swój wybór.
8. Na podstawie wcześniejszych kroków przygotuj listę słów kluczowych powiązanych z Twoim zapytaniem.
9. Przeprowadź wstępne, ręczne wyszukiwanie literatury z użyciem przygotowanych słów kluczowych w ogólnodostępnych repozytoriach open-access. Pamiętaj, że zazwyczaj nie wolno przysyłać do modeli AI artykułów objętych paywallem. Wybierz kilka (3–7) artykułów, które wydają się najbardziej istotne.
10. Przy użyciu wybranego narzędzia (ChatGPT, Scholar GPT, Perplexity lub innego zaakceptowanego przez prowadzącego) przygotuj krótkie podsumowanie artykułów wybranych w zadaniu 9.
11. Podsumuj wybrany artykuł przy pomocy narzędzia <https://notebooklm.google>. Możesz także zaprezentować jego treść w formie podcastu.
12. Na podstawie wcześniejszych analiz przygotuj rozdział przeglądu literatury dotyczący wybranego tematu, korzystając z wybranego narzędzia (ChatGPT, Scholar GPT, Perplexity lub innego zaakceptowanego przez prowadzącego). Powinien to być zwięzły, lecz wyczerpujący wstęp o objętości od 1 do 4 stron.
13. Stwórz grafiki ilustrujące główne wnioski lub najciekawsze/kluczowe zagadnienia, które warto umieścić w raporcie z zadania 12. Użyj w tym celu narzędzia <https://napkin.ai>.
14. Przygotuj prezentację podsumowującą Twoje ustalenia, ze szczególnym uwzględnieniem treści zawartych w raporcie z zadania 12. W tym celu skorzystaj z narzędzia <https://gamma.app>.

Temat 5: Demonstrator “Vanilla GAN”

1. Wybierz temat generacji danych obrazowych. Najlepiej, jeśli będą to Twoje własne fotografie (np. zdjęcia Twojej twarzy, pupila, bliskiej osoby, ulubionego roweru itd.). Musisz przygotować co najmniej sto takich obrazów.
2. Zaimplementuj podstawową sieć GAN korzystając z samouczka Matlaba: <https://www.mathworks.com/help/deeplearning/ug/train-generative-adversarial-network.html>
3. Wytrenuj swój model do generacji danych zgodnych z wybranym tematem. Pamiętaj, aby zacząć od niskiej rozdzielczości obrazów (np. 32x32 piksele). Obserwuj, ile trwa trening i jak wyglądają generowane obrazy na różnych etapach uczenia. Należy spodziewać się raczej długich czasów treningu – warto uruchomić model z domyślnymi parametrami na całą noc.
4. Oceń wpływ zmian kilku wybranych metaparametrów treningowych. Zwróć uwagę, czy któraś ze zmian wywołuje istotną różnicę w przebiegu uczenia.
5. Przygotuj raport prezentujący Twoje wyniki.
6. Oceń wpływ: (a) rozmiaru sieci (czy głębokość generatora czy dyskryminatora jest ważniejsza?) (b) wag przypisanych do treningu generatora i dyskryminatora (na który z nich warto położyć większy nacisk?)
7. Zajrzyj do literatury naukowej i wybierz jedną zmianę, która Twoim zdaniem mogłaby pozytywnie wpłynąć na jakość treningu (np. poprawa rozdzielczości generowanych danych lub lepsze odwzorowanie przykładów treningowych) – a następnie zaimplementuj tę zmianę w swoim modelu (możesz w tym zadaniu korzystać z pomocy LLM).
8. Jeśli realizowałeś zadanie 6 lub 7 – uwzględnij ich opis oraz omówienie wyników w swoim raporcie.

Temat 6: Gra w ASCII kodowana z użyciem LLM

Zadanie to dotyczy tworzenia zaawansowanego (strukturalnie) programu z wykorzystaniem wyłącznie dużych modeli językowych. Przez cały czas realizacji zadania **nie wolno Ci samodzielnie pisać żadnego kodu** – poza nadawaniem nazw plikom i funkcjom. Zapisuj swoje interakcje z modelem LLM w sposób umożliwiający ich pokazanie prowadzącemu na żądanie. **Nie wszystkie zadania oznaczone na zielono są konieczne do uzyskania oceny 5.0** – dwa dobrze zrealizowane zielone zadania mogą wystarczyć do uzyskania maksymalnej oceny.

1. Wybierz temat swojej gry. Powinien on umożliwiać projektowanie złożonych mechanik bez konieczności użycia rozbudowanych zasobów graficznych. Dobrymi przykładami są gry roguelike, tekstowe RPG lub ascii tower defence.
2. Wybierz język programowania, którego chcesz użyć. Zalecany wyborem jest Matlab – głównie dlatego, że nie „służy” on do tworzenia gier, więc ciekawe będzie sprawdzenie, jak poradzi sobie z tym zadaniem LLM. Jeśli chcesz użyć bardziej standardowego języka – możesz sięgnąć po Python, C++ lub Javę, ale pamiętaj: powinieneś znać dany język i środowisko lub być gotowym szybko nauczyć się definiowania funkcji oraz uruchamiania i debugowania kodu.
3. Wybierz swojego „towarzysza kodowania” – model LLM znany z dobrych wyników w generowaniu kodu.
4. Zastanów się nad ogólną strukturą gry (czy opiera się na zadaniach? mapach? Jakie będą główne elementy organizacyjne kodu – klasy? główne funkcje?). Konsultuj ten

- etap z wybranym modelem LLM (proces twórczy może być częściowo delegowany). Struktura gry powinna być modułowa i skalowalna – w trakcie realizacji zadania będziesz testować granice możliwości modelu poprzez rozszerzanie i komplikowanie projektu. Pomyśl, jak planujesz testować te granice.
5. Skonsultuj swoje decyzje (temat gry, język, wybrany LLM, ogólną strukturę oraz plan testu skalowalności) z prowadzącym podczas 11. laboratorium. Na tej podstawie przygotuj dokument „workplan” zawierający Twoje cele i ograniczenia.
 6. Przygotuj „wczesną wersję alfa” gry – w celu przetestowania przepływu pracy oraz sposobu komunikacji z modelem. Przeanalizuj przyjętą strukturę klas i funkcji oraz sposób ich komunikacji. Możesz to również zrobić w porozumieniu z LLM. Zapisz swoje wnioski w formie krótkiego raportu (co poszło dobrze, co nie zadziałało?).
 7. Zbuduj dopracowaną strukturę kodu – funkcja po funkcji – dbając o to, aby model pozostał skupiony na bieżącym zadaniu i zachował zgodność z planowaną strukturą kodu. Ukończ zadanie, uruchom testy beta i popraw wszystkie błędy (również wyłącznie z pomocą LLM).
 8. Przetestuj granicę złożoności strukturalnej, którą model jest w stanie utrzymać. Dodawaj mapy, postacie, przeciwników, funkcje, waluty, mechaniki itd. (zależnie od tematu gry), aż dalsze rozwijanie kodu przy użyciu wyłącznie LLM okaże się niemożliwe. Wykonanie tego zadania jest wymagane do uzyskania oceny 3.0, natomiast wyższe oceny będą wymagały większego zaangażowania.
 9. Opisz końcowy stan gry (działającą wersję), jej funkcjonalności oraz wnioski z realizacji zadania w formie raportu.
 10. Zaprojektuj i zaimplementuj nową funkcjonalność, która nie była planowana w punkcie 4 ani nie została zrealizowana w punkcie 7. Funkcjonalność ta powinna wymagać modyfikacji innych elementów kodu (np. nowy typ przeciwnika, nowy system postaci lub zdarzeń). Najpierw poproś model, by sam „zrozumiał” kod i zaproponował zmiany, a jeśli to się nie uda – przeanalizuj kod ręcznie i zleć modelowi konkretne zmiany.
 11. Wykorzystaj model do testów balansu gry. Celem jest zapewnienie dobrego doświadczenia gracza – poprzez satysfakcjonujący poziom wyzwania. Porozmawiaj z modelem o tym, które elementy gry należy zbalansować, a następnie zaprojektuj z jego pomocą matematyczny model określający ten balans. Zatwierdź listę zmian i poproś model o ich implementację we wszystkich odpowiednich miejscach gry.
 12. Dodaj zasoby do gry – niech kod korzysta z biblioteki grafik, dźwięków lub animacji. Pamiętaj, że może to wymagać zmian w „silniku gry”.
 13. Przenieś finalną wersję gry do innego języka programowania z pomocą LLM. Sprawdź, jak dużo musisz ingerować w ten proces – i co się stanie, jeśli po prostu przekażesz pliki jeden po drugim. Czy można w łatwy sposób zbudować grę w innym języku?
 14. Wykorzystaj model do stworzenia instrukcji do gry – wyjaśniającej użytkownikowi, jak ją uruchomić oraz jaka strategia prowadzi do osiągnięcia dobrego wyniku. Sprawdź jak model poradzi sobie z opisem wyłącznie na podstawie kontekstu z konwersacji w punktach 1 – 13 a następnie wejdź w interakcję z modelem poprawiając razem z nim instrukcję zgodnie ze swoją koncepcją.
 15. Przygotuj drugą część raportu rozpoczętego w punkcie 9 – tym razem podsumowującą Twoje wnioski i obserwacje z realizacji punktów 10–14.

Temat 7: Klasyfikacja rzeczywistych sygnałów w dziedzinie czasu

Temat ten wymaga odrobiny kodowania, którego nie omawialiśmy podczas zajęć – jednak możesz znaleźć gotowe samouczki realizujące podobne zadania albo skorzystać z pomocy ChatGPT przy implementacji części związanej z uczeniem głębokim.

1. Pobierz zbiór danych łożysk dostępny pod adresem: <https://engineering.case.edu/bearingdatacenter/download-data-file>. Przyjrzyj się danym. Ponieważ sygnały są długie, będziesz musiał je pociąć na fragmenty, tak aby każdy fragment stanowił „próbkę” do klasyfikacji. Zdecyduj, jak długi powinien być taki fragment. Im dłuższy, tym mniej próbek będziesz mieć do dyspozycji, ale za to cechy wyliczane z nich będą bardziej spójne. Przygotuj taki zbiór danych i podziel próbki na zbiory: treningowy, walidacyjny i testowy.
2. Zastosuj podejście oparte na płytkim uczeniu, wykorzystując proste cechy, które wyliczaliśmy na zajęciach. Sprawdź, jaką dokładność osiąga wybrany przez Ciebie model na zbiorze testowym.
3. Przejrzyj literaturę w poszukiwaniu bardziej zaawansowanych cech sygnałowych lub przeprowadź bardziej szczegółowe przetwarzanie sygnału (np. obliczając widmo częstotliwościowe), aż znajdziesz cechy, które powinny dobrze działać w Twoim przypadku. Zaimplementuj co najmniej dwie bardziej zaawansowane cechy i sprawdź, czy poprawiają one dokładność klasyfikacji.
4. Porównaj oba podejścia w formie raportu.
5. Skonfiguruj konwolucyjną sieć neuronową do klasyfikacji sygnałów. Będziesz musiał(a) zmodyfikować co najmniej: (a) sposób wczytywania danych – zamiast *imageDatastore* możesz użyć *signalDatastore* lub przygotować dane w postaci struktury *cell* z sygnałami, (b) zamienić kwadratowe kernele konwolucyjne na liniowe, jednowymiarowe, (c) zmienić warstwę wejściową z *imageInputLayer* na *sequenceInputLayer*.
Reszta kodu może działać podobnie. Użyj takiej struktury sieci, która – Twoim zdaniem – będzie odpowiednia do rozwiązania tego zadania.
6. Wytrenuj i oceń działanie swojej sieci głębokiej – w zakresie klasyfikacji różnych typów uszkodzeń sygnału. Zapisz wyniki i przygotuj ich reprezentację graficzną.
7. Przeprowadź optymalizację struktury sieci (w tym liczby warstw i rozmiarów filtrów) i oceń wyniki statystycznie.
8. Przyjrzyj się ponownie danym. Zastanów się, jakie mogą być możliwe wyjaśnienia dla uzyskanych rezultatów. Sprawdź, czy różne podziały danych dają bardziej lub mniej wiarygodne wyniki. Znajdź informacje o tym zbiorze danych w literaturze i spróbuj porównać swoje rozwiązanie z wynikami raportowanymi w pracach naukowych. Przygotuj raport opisujący podjęte kroki (w tym podejście do implementacji rozwiązania) oraz dyskusję na temat uzyskanych wyników.

Temat 8: Praktyczny klasyfikator obrazów oparty na danych własnych

1. Wybierz praktyczny problem klasyfikacyjny, który chcesz rozwiązać w oparciu o własnoręcznie pozyskany zbiór danych. W trakcie realizacji tego mini-projektu będziesz musiał przygotować dużą bazę zdjęć, na podstawie której będą trenowane Twoje klasyfikatory. Możliwe przykłady to: klasyfikacja drzew na podstawie liści, klasyfikacja potraw na podstawie zdjęcia z góry, klasyfikacja znaków drogowych na kategorie. Weź pod uwagę ograniczenia etyczne – wybierz temat, dla którego będziesz mógł wykonać setki zdjęć. **Skonsultuj temat z prowadzącym i uzyskaj jego akceptację.**
2. Zaplanuj część eksperymentalną. Przemyśl, ile zdjęć należy pozyskać, w jakich warunkach, kto będzie je pozyskiwał i jak zapewnisz, że tworzony zbiór danych będzie wystarczająco reprezentatywny, by umożliwić trening wiarygodnych klasyfikatorów. **Skonsultuj plan z prowadzącym i uzyskaj jego akceptację.** Przygotuj dokumentację z planem części eksperymentalnej.
3. Poproś prowadzącego o uzupełnienie planu o projekt „*dodatkowego zbioru testowego*” (prowadzący zmodyfikuje Twój plan eksperymentu, dodając projekt innego doświadczenia, które posłuży jako dane testowe). Uwzględnij go w swojej dokumentacji.
4. Pozyskaj dane ściśle według przygotowanego planu.
5. Pozyskaj dodatkowe dane zgodnie z planem dostarczonym przez prowadzącego.
6. Jeśli uznasz, że coś istotnego zostało pominięte w Twoim pierwotnym planie lub zauważysz czynniki ryzyka i chcesz ostatecznie mieć więcej danych – możesz pozyskać dane dodatkowe w celu dalszego testowania i uzupełnienia zbioru. Oddziel je jednak wyraźnie od danych z eksperymentu podstawowego (pkt 4 i 5).
7. Oznacz swój zbiór danych i przygotuj dokument z opisem eksperymentu.
8. Podziel dane losowo na zbiory treningowy, walidacyjny i testowy, wyłączając z nich zestaw nauczyciela oraz inne zestawy pomocnicze – nazwijmy ten podział **Split 1**.
9. Podziel dane na zbiory treningowy, walidacyjny i testowy w sposób świadomy – np. w oparciu o czas pozyskania danych, osobę wykonującą zdjęcia, miejsce, użyty sprzęt itp., tak aby zapewnić niezależność między zbiorami – nazwijmy ten podział **Split 2**.
10. Przeprowadź standardową optymalizację metaparametrów oraz trening klasyfikatora (zgodnie z instrukcją nr 11) na zbiorach treningowym i walidacyjnym, a następnie przetestuj go na zbiorze testowym – dla Split 1.
11. Powtórz punkt 10 dla Split 2.
12. Porównaj statystycznie (na podstawie wielu treningów) skuteczność klasyfikatora na Twoim zbiorze testowym oraz na zbiorze testowym dostarczonym przez prowadzącego. Jeśli posiadasz Split 2 – również go przetestuj.
13. Porównaj i omów różne podejścia do trenowania i konfiguracji klasyfikatora (bądź przygotowany na zaprezentowanie wyników kolegom i prowadzącemu w formie graficznej lub tabelarycznej).
14. Złóż raport zawierający dane, wyniki eksperymentów oraz opis eksperymentu.