

SZCZEGÓŁOWY PLAN SZKOLEŃ

I ETAP SZKOLEŃ – wykłady online [około 24h]

Elementy rachunku prawdopodobieństwa i statystyki matematycznej

wykładowca: Hanna Wojewódka-Ściążko

- Modelowanie doświadczenia losowego za pomocą **przestrzeni probablistycznej**.
- **Zmienna losowa** - intuicje, przykłady, formalna definicja.
- Typowe rozkłady prawdopodobieństwa (w tym **rozkład normalny**).
- Zmienne losowe niezależne.
- Centralne twierdzenie graniczne.
- Rozkład t Studenta.
- Teoretyczne podstawy wnioskowania statystycznego.
- Weryfikacja hipotez statystycznych:
 - parametryczne testy istotności (przykład - **test t**);
 - testy różnicy średnich dla obserwacji powiązanych w pary (przykłady – **sparowany test t i test Wilcoxon**);
 - testy zgodności, testy normalności rozkładu (przykład - **test normalności Shapiro-Wilka**).

Podstawy uczenia maszynowego (ML)

wykładowca: Przemysław Głomb

- Podstawowe zagadnienia ML - cele, pojęcia, zadania, wprowadzenie.
- Przykładowe dane (klasyfikacja hiperspektralna krwi).
- **Sieci neuronowe** - zasada działania, proces uczenia, selekcja i weryfikacja modeli.
- **Metody jądrowe** - wprowadzenie, **klasyfikator SVM**.
- Zaawansowane sieci neuronowe.
- “Skrzynka narzędziowa” uczenia maszynowego.

Elementy algebry liniowej

wykładowca: Hanna Wojewódka-Ściążko

- **Macierze** i działania na macierzach
- Wektory i wartości własne macierzy.
- **Przestrzeń wektorowa**.
- Iloczyn skalarny.
- Notacja Diraca i pojęcie **bitu kwantowego** (kubitu).

Algorytmy numeryczne znajdujące minimum lokalne zadanej funkcji celu

wykładowca: [Hanna Wojewódka-Ściążko](#)

- Pojęcie **gradientu**.
- Metoda spadku wzdłuż gradientu.
- **Metoda stochastycznego spadku wzdłuż gradientu**.

Wprowadzenie do obliczeń kwantowych

wykładowca: [Zbigniew Puchała](#)

- Zagadnienia wstępne, fizyka mikroświata, efekty kwantowe.
- Ewolucja układu w czasie. **Równanie Schroedingera**.
- **Stany kwantowe i superpozycja stanów** – definicje i przykłady.
- **Pomiar kwantowy** – czym jest i jakie są jego efekty?
- Hamiltonian, ewolucja unitarna.
- Bramki kwantowe.
- Układy złożone, iloczyn tensorowy, stany produktowe, **stany splątane, stany Bella**.
- **Informacja klasyczna vs informacja kwantowa**. Częściowa konwersja obu form informacji w siebie.
- *No cloning theorem*.
- **Kwantowe gęste kodowanie**. Kwantowa teleportacja.
- Kwantowa kryptografia.

Algorytmy kwantowe

wykładowca: [Zbigniew Puchała](#)

- **Algorytm Shore'a**: badanie okresowości funkcji (przykład algorytmu faktoryzacji)
- Algorytm kwantowy Deutsch-Jozsa.
- **Algorytm Grovera**: poszukiwanie elementu znaczonego.
- Twierdzenie Shannona – kwantowa informacja, macierze gęstości.
- Kompresja Schumachera.
- **Kwantowa korekcja błędów**.

II ETAP SZKOLEŃ – warsztaty w kameralnych grupach (prowadzone stacjonarnie) [około 24h]

Uczenie maszynowe – część I (*data mining*)

tutor: Przemysław Głomb

- Analiza i przygotowanie zbiorów danych:
 - podstawowe **narzędzia** *data mining* (statystyki opisowe, algorytmy grupowania, algorytmy rzutowania),
 - zasady ich użycia,
 - ćwiczenia praktyczne z przygotowywania zbiorów dla klasyfikacji i regresji.

Uczenie maszynowe – część II (klasyfikacja)

tutor: Przemysław Głomb

- Budowa eksperymentu klasyfikacji:
 - **miary wydajności**,
 - podział zbiorów (walidacja krzyżowa, optymalizacja hiperparametrów, przetrenowanie, przekleństwo wymiarowości),
 - porównanie klasyfikatorów testami statystycznymi,
 - wybór metod przetwarzania wstępnego.
- Najważniejsze **klasyfikatory**. Konfigurowanie i wykorzystanie SVM. Konfiguracja i trenowanie sieci MLP (krzywa loss, optymalizacja, funkcje aktywacji, hiperparametry).

Uczenie maszynowe – część III (sztuczne sieci neuronowe i sieci głębokiego uczenia)

tutor: Przemysław Głomb

- **Diagnostyka procesu treningu sieci neuronowej**, weryfikacja skuteczności, ataki na sieć, wyjaśnianie działania sieci.
- Zaawansowane **architektury sieci neuronowych**, głębokie hierarchie cech, implementacja treningu (w tym *transfer learning*).

Uczenie architektur kwantowych – część I (wprowadzenie)

tutor: Łukasz Paweła

- **Bramkowe komputery kwantowe**: Szczegółowy opis budowy i zasad działania komputerów kwantowych opartych na bramkach.
- Kubity i stany kubitu: Omówienie podstawowych jednostek informacji kwantowej, ich reprezentacji i właściwości na architekturach IBMQ.
- Splątanie kwantowe: Wyjaśnienie fenomenu splątania oraz jego znaczenia w obliczeniach kwantowych na architekturach IBMQ.
- Bramkowy model obliczeń: Przedstawienie zasad działania komputerów kwantowych opartych na bramkach w odniesieniu do architektur IBMQ.

- Obwody kwantowe: Budowa i analiza podstawowych obwodów kwantowych.
- Perceptron: Przypomnienie klasycznego perceptronu oraz zasad jego działania i uczenia.
- Python: Krótkie przypomnienie podstawowej składni języka Python, niezbędnej do implementacji obwodów kwantowych.
- Ograniczenia komputerów kwantowych: Omówienie problemów i ograniczeń związanych z obecnym stanem technologii kwantowej.
- **Źródła błędów w komputerach bramkowych:** Analiza typowych źródeł błędów w obliczeniach kwantowych i ich wpływu na wyniki.
- **Kwantowy model perceptronu:** Budowa, elementy składowe oraz przykładowe obwody kwantowego perceptronu.

Uczenie architektur kwantowych – część II (pierwsze implementacje)

tutor: Łukasz Paweła

- Uzyskanie dostępu do **IBMQ**: Proces rejestracji i uzyskania dostępu do platformy IBMQ.
- Podstawy biblioteki Qiskit: Wprowadzenie do biblioteki Qiskit, narzędzia do programowania komputerów kwantowych.
- Podstawowe operacje w portalu IBMQ: Nauka monitorowania i zarządzania obliczeniami kwantowymi za pomocą portalu IBMQ.
- Symulatory architektur komputerów bramkowych: Implementacja pierwszego programu na symulatorze kwantowym.
- Symulatory lokalne vs chmurowe: Porównanie lokalnych i chmurowych symulatorów kwantowych.
- Implementacja programu uruchamianego na komputerze bramkowym: Praktyczna realizacja przygotowania stanu splątanego.
- **Implementacja składowych perceptronu:** Tworzenie i testowanie podstawowych elementów kwantowego perceptronu.

Uczenie architektur kwantowych – część III (programowanie)

tutor: Łukasz Paweła

- Obserwacje działania modelu dla losowo dobranych wag: Analiza wyników działania perceptronu z losowo wybranymi wagami.
- Implementacja algorytmu uczenia kwantowego perceptronu: Kodowanie i testowanie algorytmu uczącego perceptronu.
- **Uczenie kwantowego modelu perceptronu**, weryfikacja wyników: Proces uczenia modelu perceptronu i ocena jego skuteczności.
- Model bramkowy vs wyżarzanie kwantowe: Porównanie dwóch podejść do obliczeń kwantowych: modelu bramkowego i wyżarzania kwantowego.
- **Algorytm QAOA:** Przegląd algorytmu Quantum Approximate Optimization Algorithm (QAOA) i jego zastosowań.

Zagadnienia dodatkowe:

- Model bramkowy vs wyżarzanie kwantowe.
- Algorytm QAOA.

III ETAP SZKOLEŃ – warsztaty w kameralnych grupach (prowadzone stacjonarnie) [około 24h]

Kwantowe sieci neuronowe i kwantowe metody jądrowe – część I (bramki i obwody parametryzowalne)

tutor: Piotr Gawron

- Kwantowe bramki parametryzowalne.
- Kwantowe obwody parametryzowalne.
- Kwantowe algorytmy wariacyjne.
- Różniczkowanie funkcji opartych o obwody parametryzowalne.
- Kwantowa metoda przesuwania parametrów.
- **Optymalizacja parametrów obwodów kwantowych.**

Kwantowe sieci neuronowe i kwantowe metody jądrowe – część II (klasyfikacja)

tutor: Piotr Gawron

- **Kwantowe sieci neuronowe.**
- Ładowanie danych do sieci.
- Architektury sieci kwantowych.
- **Kwantowe metody jądrowe.**
- Klasyfikacja za pomocą kwantowych sieci neuronowych i kwantowych metod jądrowych.

Zagadnienia dodatkowe:

- Łączenie sieci kwantowych i klasycznych sieci neuronowych.
- Trudności związane z uczeniem i wykorzystywanie kwantowych metod uczenia maszynowego.

Kwantowe sieci neuronowe i kwantowe metody jądrowe – część III (programowanie)

tutor: Piotr Gawron

- Biblioteka PennyLane.
- Implementacja prostego klasyfikatora jedno-kubitowego.
- Implementacja klasyfikatora z ponownym ładowaniem danych.
- **Implementacja klasyfikatora wykorzystującego splątanie.**
- Dobór architektury sieci.
- **Implementacja klasyfikatora SVM wykorzystującego kwantową metodę jądrową.**

Kwantowe wyżarzanie kombinatorycznych problemów optymalizacyjnych – część I (klasyczne modele QUBO i Isinga)

tutor: [Bartłomiej Gardas](#)

- Klasyczny **model Isinga** – znaczenie, kodowanie, stany nisko-energetyczne.
- Interpretacja fizyczna modelu *Isinga* – znaczenie oddziaływań i pól magnetycznych.
- **Model QUBO** – znaczenie, przykłady, równoważność z modelem Isinga.
- Kodowanie dyskretnych problemów optymalizacyjnych za pomocą QUBO/ Isinga.
- Algorytm wyczerpującego przeszukiwania (*Brute-Force*) dla modelu QUBO/ Isinga – certyfikacja, ograniczenia.
- Analiza złożoności problemu poszukiwania stanu podstawowego – trudności i wyzwania.

Kwantowe wyżarzanie kombinatorycznych problemów optymalizacyjnych – część I (twierdzenie adiabatyczne)

tutor: [Bartłomiej Gardas](#)

- Podejście heurystyczne – przegląd klasycznych algorytmów inspirowanych fizycznie.
- Kwantowanie modelu Isinga – **kwantowy model Isinga** z poprzecznym polem.
- Związek pomiędzy klasycznym a kwantowym modelem Isinga – problem własny (macierze zagadnienie własne) a sortowanie.
- **Twierdzenie adiabatyczne** w mechanice kwantowej oraz **kwantowe wyżarzanie**.
- **Kwantowy procesor wyżarzający D-Wave** – opis, dostępne topologie, osadzanie problemów.

Kwantowe wyżarzanie kombinatorycznych problemów optymalizacyjnych – część III (programowanie)

tutor: [Bartłomiej Gardas](#)

- Rozwiązywanie modelu Isinga z wykorzystaniem kwantowego wyżarzania.
- Praktyczne wykorzystanie biblioteki *D-Wave Ocean*.
- Implementacja algorytmu wyczerpującego (*Brute-Force*).
- Implementacja algorytmu równoległego wyżarzania.
- Implementacja algorytmu typu *Branch & Bound*.
- **Implementacja algorytmu symulowanej bifurkacji.**
- **Implementacja algorytmu symulowanego wyżarzania.**
- Wykorzystanie procesorów graficznych (GPU) w algorytmach heurystycznych.
- Analiza porównawcza algorytmów dla wybranych instancji problemów Isinga.

Zagadnienia dodatkowe:

- Sieci tensorowe – rodzaje, konstrukcja, zwężanie.
- Sieć tensorowa typu MPS (1D) oraz PEPS (2D).
- Emulacja procesorów wyżarzających za pomocą sieci tensorowych.
- Różnorodność stanów nisko-energetycznych, droplety oraz ich znaczenie.