

Technologie kwantowe w uczeniu maszynowym

dr Sebastian Zajac

Seminarium
Wojskowa Akademia Techniczna

17.10.2024

Sztuczna inteligencja

Rozwój związany z przetwarzaniem danych wspomagający rozpoznawanie wzorców (ML, DL), ale również generowanie i tworzenie nowych danych (GenAi, LLM).

klasyczny model AI

$$f(x; \theta)$$

- Wykładniczy wzrost ilości (treningowych) danych dostępnych w social mediach, internecie, aplikacjach mobilnych, IoT.
- Wzrost możliwości i spadek ceny (kosztów) jakości sprzętu komputerowego - co-procesory GPU, TPU, ...
- Oprogramowanie Open Source

Sztuczna inteligencja

Historia AI od ok 1950

Rozwój technologii związanych z danymi ustrukturyzowanymi i nieustrukturyzowanymi realizowany np. w:

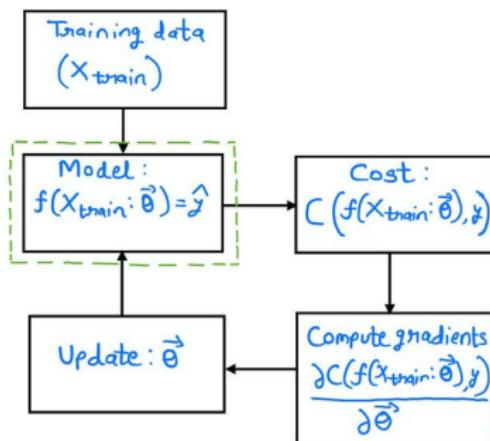
- rozpoznawanie i tworzenie „mowy”
- „rozumienie” obrazów i filmów
- rozgrywanie gier
- rekomendacje
- systemy diagnostyczne
- planowanie
- podejmowanie decyzji
- boty, agenci ...

Zadania te realizowane są w oparciu o ML i DL.

Machine Learning, Deep Learning

ML jak przypisać nową wartość dla nowej kombinacji wartości zmiennych.
Możemy zrealizować:

- **klasyfikacja** - przewidywanie wartości dyskretnej,
- **regresja** - przewidywanie wartości ciągłej,



segmentacja (podział populacji na grupy).

Deep Learning

THE NOBEL PRIZE

Nobel Prizes & laureates About Stories Educational Events & museums



John J. Hopfield

The Nobel Prize in Physics 2024

Prize motivation: "for foundational discoveries and inventions that enable machine learning with artificial neural networks"



Geoffrey E. Hinton

The Nobel Prize in Physics 2024

Prize motivation: "for foundational discoveries and inventions that enable machine learning with artificial neural networks"

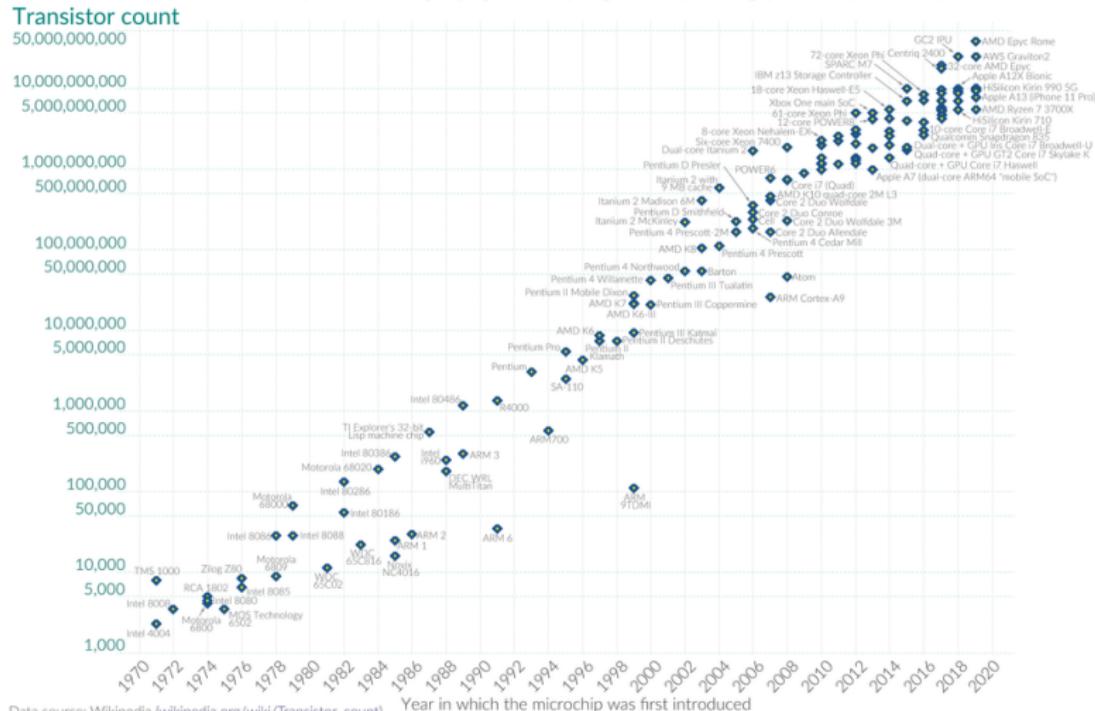


Klasyczne komputery

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World
in Data



Data source: Wikipedia ([wikipedia.org/wiki/Transistor_count](https://en.wikipedia.org/wiki/Transistor_count))

OurWorldInData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Klasyczne komputery

Zwiększenie szybkości działania oraz pojemności klasycznych komputerów co dwa lata zmniejszając przy tym koszty.

Fizyczne (klasyczne) ograniczenia

- Rozmiary tranzystora > rozmiary atomów lata 90 500nm obecnie 14nm i 7nm
- Prędkość światła \sim max. prędkość przesyłu informacji.
- wysoki koszt wytwarzania (kierunek układów wieloprocesorowych)
- wysoki pobór prądu, ciepło

Richard Feynman

Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy.

- 1936 Alan Turing "On Computable Numbers"
- 1976 Roman S. Ingarden "Quantum Information Theory"
- 1980 Paul Benioff - teoretyczna możliwość komputerów kwantowych
- 1981 Richard Feynman - symulowanie procesów kwantowych.
- 1994 Peter Shor - Algorytm faktoryzacji liczb w czasie wielomianowym.
- 1996 Lov Grover - Algorytm Grover'a
- 2000 pierwszy 5-kubiotwy kwantowy komputer
- 2011 D-Wave pierwsza sprzedaż komercyjnego komputera kwantowego
- 2019 quantum supremacy
- ...

Quantum Computing

Obliczenia kwantowe to nowy paradymat przetwarzania informacji wykorzystujący własności mechaniki kwantowej - *kubit, superpozycja, splątanie*.

Modele obliczeń kwantowych

- Quantum Circuits - bramkowy model obliczeń kwantowych
- adiabatyczne obliczenia kwantowe,
- topologiczne komputery kwantowe

jeszcze jedna zasada

Dekoherencja - oddziaływanie „niszczące” stan kwantowy
ERA Noisy Intermediate-Scale Quantum (NISQ)

The Nobel Prize in Physics 2022



III. Niklas Elmehed © Nobel Prize Outreach
Alain Aspect



III. Niklas Elmehed © Nobel Prize Outreach
John F. Clauser



III. Niklas Elmehed © Nobel Prize Outreach
Anton Zeilinger

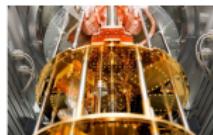
for experiments with entangled photons, establishing the violation of Bell inequalities and pioneering quantum information science"

(2021-2023)

Quantum Machines



Google
(Superconducting)



Baidu Qianshi
(Superconducting)

ATOS
(Simulator)



Rigetti
(Superconducting)



IBM (Superconducting)

Xanadu (Photonic)



What makes them all work?

Qubit (or quantum bit) is the fundamental model of quantum information and its manipulation

Quantum circuit is a model of quantum computation, involving qubits and operations on them



4 / 17

Kwantowe Bity

CLASSICAL COMPUTERS



1 STATE AT A TIME

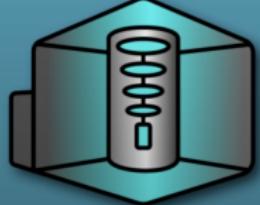
BITS



BITS ARE INDEPENDENT OF EACH OTHER

CLASSICAL VS. QUANTUM

QUANTUM COMPUTERS



SUPERPOSITION
ENTANGLEMENT
INTERFERENCE

MANY STATES AT A TIME

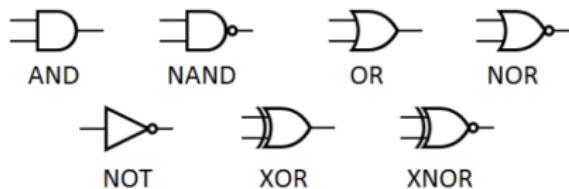
QUBITS



QUBITS ARE IN A COMBINED STATE TOGETHER

Bramki kwantowe

Klasyczne bramki kwantowe działają na bitach i są najczęściej nieodwracalne. Klasyczne bramki używane najczęściej:



Kwantowe bramki (realizują operatory liniowe), są w pełni odwracalne.

$$U = e^{-iHt}$$

takie, że: $UU^\dagger = U^\dagger U = I$. Wiele bramek kwantowych nie ma odpowiedników klasycznych: np. Bramka Hadamarda, Bramka CNOT.

Splątanie kwantowe

Splątanie kwantowe to własność złożonych układów kwantowych.

Klasycznie operujemy na bitach połączonych w bajty.

Kubity możemy łączyć w tzw. rejesty kwantowe. Jeden kubit może istnieć w superpozycji 2 stanów. N kubitów to superpozycja 2^n stanów.

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \quad \text{and} \quad |\phi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$|\psi\rangle \otimes |\phi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \otimes \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix} = \alpha_0\beta_0|0\rangle \otimes |0\rangle + \alpha_0\beta_1|0\rangle \otimes |1\rangle + \alpha_1\beta_0|1\rangle \otimes |0\rangle + \alpha_1\beta_1|1\rangle \otimes |1\rangle$$

Przykład 3 kubity:

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle \equiv |000\rangle$$

$$|\psi_1\rangle = |0\rangle \otimes |0\rangle \otimes |1\rangle \equiv |001\rangle$$

⋮

$$|\psi_7\rangle = |1\rangle \otimes |1\rangle \otimes |1\rangle \equiv |111\rangle$$

Stan splątany

dwa kubity - iloczyn tensorowy

Weźmy dwa kubity i zapiszmy je jako iloczyn tensorowy:

$$|\psi_1\rangle = \gamma_0|00\rangle + \gamma_1|01\rangle + \gamma_2|10\rangle + \gamma_3|11\rangle$$

Dokonujemy pomiaru jednego kubitu.

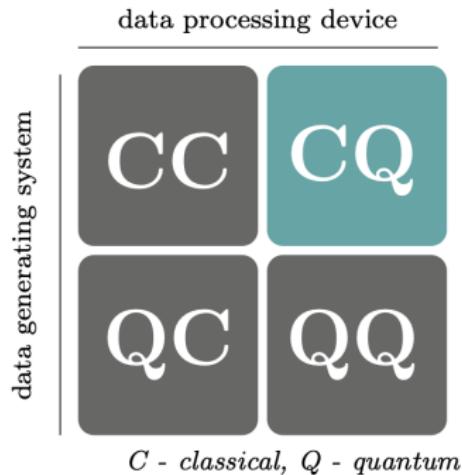
Splątane dwa kubity

Istnieją jednak stany dwóch kubitów, których nie da się przedstawić jako iloczyn tensorowy ich składników - tzw. Stany Bella:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

pomiar pierwszego kubitu definiuje stan drugiego kubitu.

Kwantowe uczenie maszynowe

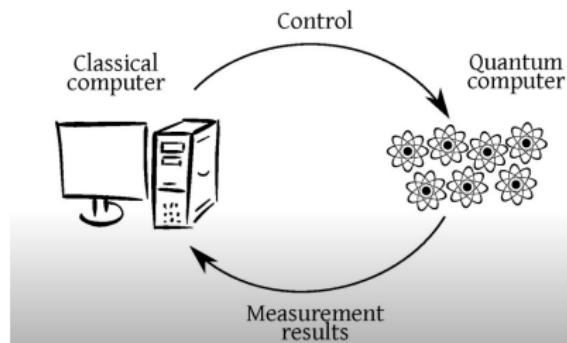


$$|f(x; \theta)\rangle$$

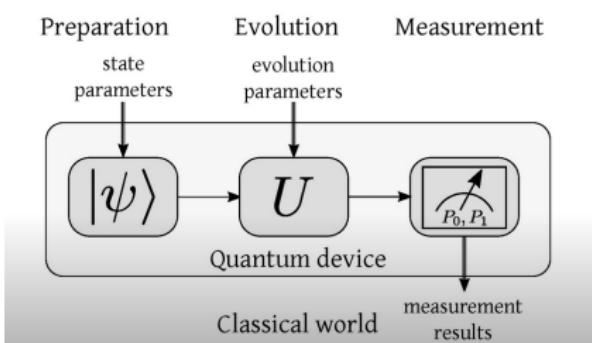
Czy możemy zamienić „model” wykorzystując obliczenia kwantowe?

Proces obliczeń kwantowych

Quantum computation control loop



Computation as experiment

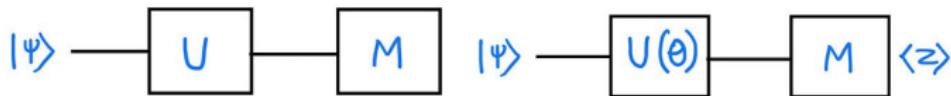


Przykład (deterministyczny): Algorytm Deutsch-Jozsa

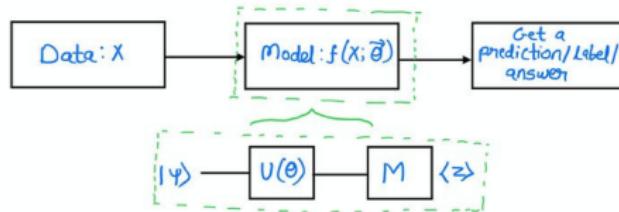
Wariacyjne modele kwantowe

Variational Quantum Circuits

Trenujmy nasze komputery kwantowe tak jak robimy to w sieciach neuronowych.



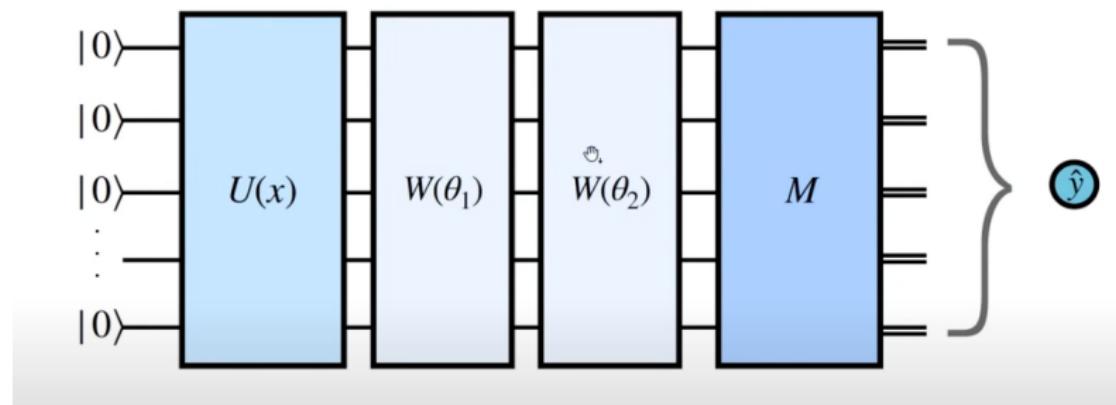
Variational Circuit as Classifier:



Variational quantum eigensolver, Variational quantum classifier, Quantum Support Vector Machine, ...

Quantum neural networks

Variational classifier



inne architektury:

Training deep quantum neural networks (Beer et al. 2020 Nature),
Variational quantum autoencoders (Romero et al. 2017 Science),
Denoising Quantum Time Series Autoencoder (J. Cybulski, S.Z 2024),
Quantum Convolutional NN (Cong et al. Nature 2019)

Kod - jeszcze nie w superpozycji

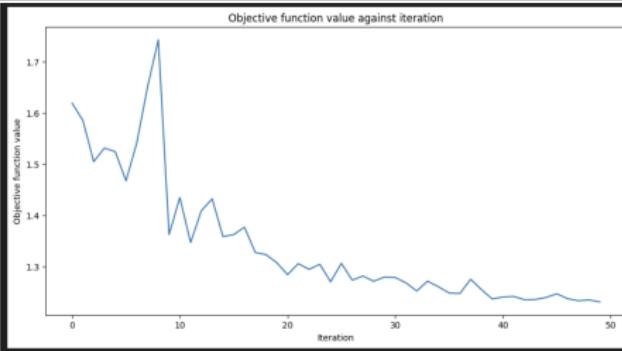
```
import time
from qiskit_machine_learning.algorithms.classifiers import VQC

vqc = VQC(
    sampler=sampler,
    feature_map=feature_map,
    ansatz=ansatz,
    optimizer=optimizer,
    callback=callback_graph,
)

# clear objective value history
objective_func_vals = []

start = time.time()
vqc.fit(X, y)
elapsed = time.time() - start

print(f"Czas uczenia: {round(elapsed)} sekund")
```



Gdzie zacząć?

Obliczenia kwantowe

- IBM Quantum Lab Jupyterlab
- D-Wave
- Xanadu
- AWS Braket
- Google Cirq

Programowanie

- Python: Qiskit, PennyLane, TensorFlow Quantum, Cirq
- Julia: Braket.jl, Yao.jl

QPoland, QuantumAI foundation, Wprowadzenie do kwantowego uczenia maszynowego- wykład SGH.