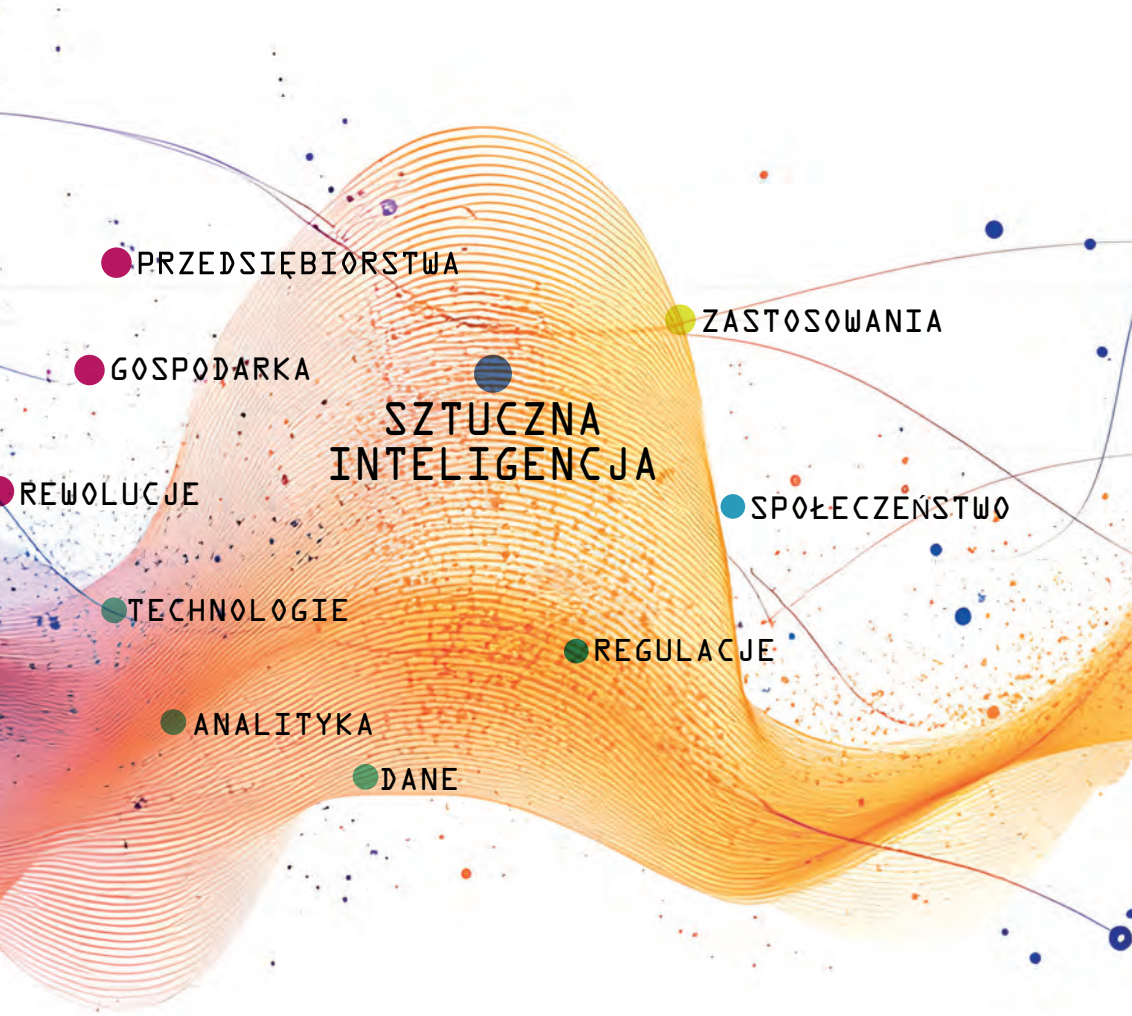
Redakcja naukowa **Tymoteusz Doligalski** • **Daniel Kaszyński**

SZTUCZNA INTELIGENCJA W PRZEDSIĘBIORSTWACH I GOSPODARCE (AI SPRING 2024)



SGH Oficyna Wydawnicza

Recenzje
Artur Strzelecki
Sylwia Sysko-Romańczuk

Redakcja językowa
Dorota Krowicka

Wydanie I

ISBN 978-83-8030-735-3

Oficyna Wydawnicza SGH – Szkoła Główna Handlowa w Warszawie
02-554 Warszawa, al. Niepodległości 162
www.wydawnictwo.sgh.waw.pl
e-mail: wydawnictwo@sgh.waw.pl

Projekt i wykonanie okładki
Magdalena Limbach

Skład i łamanie
DM Quadro

Druk i oprawa
Quick Druk

Zamówienie 72/V/25

10.0

KWANTOWE ALGORYTMY HYBRYDOWE JAKO MODELE UCZENIA MASZYNOWEGO

Sebastian Zając
Szkoła Główna Handlowa w Warszawie

Jacob Cybulski
Deakin University Melbourne

Tomasz Kulpa
Uniwersytet Kardynała Stefana Wyszyńskiego w Warszawie

Streszczenie

W rozdziale zaprezentowano kwantowe algorytmy hybrydowe. Pozwalają one realizować klasyczne modele uczenia maszynowego z wykorzystaniem obwodów kwantowych. Po wprowadzeniu podstawowych pojęć przedstawiono algorytm kwantowej sieci neuronowej, który można wykorzystać zarówno do procesu predykcji wartości ciągłej, jak i w procesie klasyfikacji.

Słowa kluczowe: obliczenia kwantowe, kwantowe uczenie maszynowe, sztuczna inteligencja

Wprowadzenie

Wraz z dynamicznym wzrostem ilości generowanych danych rośnie zapotrzebowanie na moc obliczeniową, umożliwiającą przetwarzanie tych danych w skończonym czasie, oraz wspieranie podejmowania decyzji biznesowych. Pomimo postępu technologicznego, w tym dostępu do coraz tańszego i bardziej wydajnego sprzętu, wiele problemów obliczeniowych wciąż pozostaje nierozwiązywalnych w rozsądnym czasie. Tradycyjnie wyzwania te dotyczyły głównie fizyki i chemii, jednak współczesne zastosowania biznesowe, takie jak optymalizacja i sztuczna inteligencja, również wymagają narzędzi o ogromnej mocy obliczeniowej. Obecne komputery często nie są w stanie sprostać tym wymaganiom. Technologią, która próbuje rozwiązać problemy obliczeniowe klasycznych komputerów, są komputery kwantowe, stosowane w takich dziedzinach, jak kryptografia, symulacje fizyczne i chemiczne, uczenie maszynowe i optymalizacja. Komputery kwantowe nie są jeszcze maszynami, które moglibyśmy wykorzystać do codziennych zadań. Te dostępne pozwalają wykonywać algorytmy i obliczenia na setkach kubitów. Nie posiadają jednak pełnego mechanizmu korekcji błędów. Ponadto bramki muszą działać znacznie szybciej niż ich czas dekoherencji, co uniemożliwia realizację długich sekwencji bramek dla złożonych algorytmów. Dlatego obecny etap rozwoju tych maszyn nazywany jest erą **NISQ** (*noisy intermediate-scale quantum)* [Preskill, 2018]. Pomimo ograniczeń technologicznych wciąż można wykazać tzw. kwantową supremację (czyli przewagę algorytmów kwantowych nad klasycznymi) w problemach optymalizacyjnych i w modelowaniu danych, wykorzystując do tego celu parametryzowane obwody kwantowe (*parameterized quantum circuits*, PQCs), które z zastosowaniem klasycznych optymalizatorów mogą być trenowane w celu znalezienia optymalnych wartości dla zadanej funkcji kosztu. Podejście takie nazywane jest uczeniem hybrydowym. PQC realizowane są z użyciem bramek w postaci ustalonej (np. bramki CNOT). Wykorzystują one również bramki parametryzowane, co pozwala generować nietrywialne wyniki [Lund, 2017; Harrow, 2017]. Modele kwantowego uczenia maszynowego (*quantum machine learning*, QML) realizowane przez kwantowe algorytmy wariacyjne (*variational quantum algorithms*, VQA) reprezentują całą klasę algorytmów, które używają klasycznych optymalizatorów do znalezienia parametrów kwantowych obwodów. Szczególnymi realizacjami tak zdefiniowanych modeli są: *variational quantum eigensolver* [Peruzzo, 2014], *variational quantum solvers* [Cerezo, 2021], *variational quantum classifier* [Havlicek, 2019], *quantum support vector classification* [Hsu, 2020], *quantum neural networks* [Benedetti, 2019], *quantum autoencoder* [Cybulski, 2024] czy też *quantum approximate optimization algorithm*, stosowane w zadaniach optymalizacyjnych typu QUBO [Farhi, 2014].

Wszystkie tego typu algorytmy można realizować w bibliotekach Pythona, takich jak IBM Qiskit, Pennylane, Cirq z wykorzystaniem prawdziwych komputerów kwantowych, udostępnianych przez dostawców chmurowych, typu AWS, Google Quantum AI, Azure Quantum, IBM Q Experience, Leap (D-Wave) i wielu innych.

10.1. Modele obliczeń kwantowych

Model obliczeń kwantowych znacząco różni się od klasycznego podejścia. Aby zrozumieć te różnice, wprowadźmy podstawowe pojęcia i terminologię stosowaną zarówno w obliczeniach klasycznych, jak i kwantowych.

Klasyczne komputery przetwarzają informacje zakodowane w postaci ciągu bitów. Każdy bit przyjmuje jedną z dwu wartości: 0 lub 1. Tranzystory, będące podstawowym elementem współczesnych komputerów, umożliwiają zarówno zapis informacji (utworzenie stanu początkowego), jak i jej odczyt (weryfikację stanu końcowego). Proces obliczeń realizowany jest poprzez zastosowanie bramek logicznych, które przekształcają stan początkowy bitów w stan końcowy zgodnie z zaprogramowanymi regułami.

Teoretyczny opis działania klasycznych komputerów opiera się na koncepcjach zaproponowanych przez Turinga [1950] oraz von Neumanna [1945]. Alan Turing wprowadził pojęcie maszyny Turinga jako abstrakcyjnego modelu obliczeniowego, który może wykonać dowolny algorytm, jeśli zostanie odpowiednio zaprogramowany. Natomiast John von Neumann opisał architekturę współczesnych komputerów, opartą na centralnym procesorze (*central processing unit*, CPU), pamięci oraz jednostkach wejścia – wyjścia. Szczegółowe informacje na temat działania komputerów klasycznych można znaleźć w literaturze [Wong, 2022; Feynman, 2022]. Publikacje te dostarczają wiedzy zarówno teoretycznej, jak i praktycznej na temat klasycznych modeli obliczeniowych.

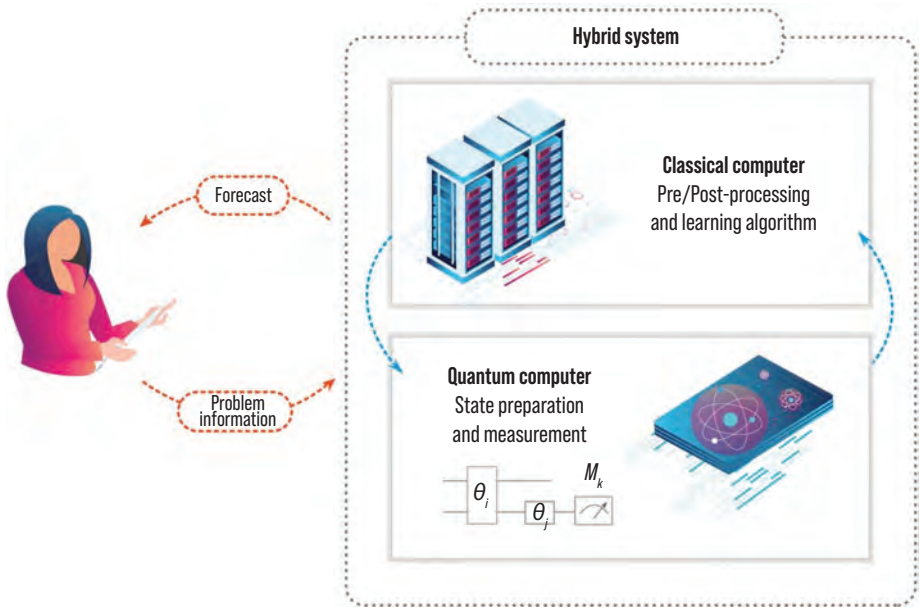
W obszarze obliczeń kwantowych wyróżnia się trzy podstawowe modele, które różnią się metodami realizacji obliczeń. Są to:

- **obwody kwantowe** (*quantum circuits*), wykorzystujące bramki kwantowe do wykonywania operacji na kubitach [Nielsen, Chuang, 2010];
- **adiabatyczne obliczenia kwantowe,** zaproponowane m.in. przez firmę D-Wave, pozwalające rozwiązywać problemy optymalizacyjne z zastosowaniem kwantowego wyżarzania [Hauke *et al.*, 2020];
- **topologiczne komputery kwantowe** rozwijane m.in. przez firmę Microsoft, bazujące na zaawansowanej topologii algebraicznej oraz oferujące innowacyjne podejście do realizacji obliczeń kwantowych [Nayak *et al.*, 2008].

W niniejszej pracy skoncentrujemy się wyłącznie na modelu opartym na obwodach kwantowych.

Pierwszy procesor kwantowy został opracowany w 1996 r. [Gershenfeld, 1996]. Urządzenie to, bazujące na technologii magnetycznego rezonansu jądrowego (*nuclear magnetic resonance*, NMR), wykorzystywało impulsy radiowe do realizacji bramek kwantowych i obsługiwało dwa kubity. W 2016 r. firma IBM wprowadziła 5-kubitowy procesor kwantowy dostępny w środowisku chmurowym. Obecnie, w 2025 r., dostępne są komputery kwantowe umożliwiające wykonanie obliczeń na ponad 1000 kubitach.

Rysunek 10.1. Schemat hybrydowego procesu obliczeń kwantowych



Źródło: Benedetti [2019].

Proces obliczeń kwantowych, zilustrowany na rysunku 10.1, wymaga współpracy klasycznego komputera z procesorem kwantowym. Klasyczny komputer inicjalizuje stan początkowy kubitów za pomocą instrukcji programistycznych, definiuje konfigurację bramek kwantowych, które realizują dany algorytm, oraz przesyła te ustawienia do procesora kwantowego. Procesor kwantowy przetwarza algorytm, modyfikując stan kubitów zgodnie z programem. Po zakończeniu przetwarzania kubity są mierzone, a wyniki pomiarów przesyłane z powrotem do klasycznego komputera, który rejestruje i analizuje dane. Proces ten można porównać do działania procesorów graficznych (*graphics processing unit*, GPU) wykorzystywanych w zadaniach, takich jak wyznacza-

nie parametrów sieci neuronowych. Dane wejściowe, przedstawione w postaci macierzy, są ładowane do karty graficznej jako bity, gdzie GPU realizuje obliczenia poprzez operacje logiczne. Po zakończeniu obliczeń wyniki są odczytywane i interpretowane przez komputer wyposażony w jednostkę centralną przetwarzania.

10.2. Definicje matematyczne wykorzystywane w obliczeniach kwantowych

Kubit (*quantum bit*) jest podstawowym nośnikiem informacji w obliczeniach kwantowych. Fizycznie może on być realizowany jako dwustanowy układ kwantowy. Stan kubitu opisujemy jako wektor w przestrzeni Hilberta. Opis ten może dotyczyć zarówno pojedynczego kubitu, jak i całego zbioru kubitów. Aby wyjaśnić kluczowe koncepcje algebry liniowej stosowanej w obliczeniach kwantowych, wykorzystujemy notację Diraca. W tej notacji wektory są reprezentowane przez symbole: bra „$\langle \cdot |$" oraz ket „$| \cdot \rangle$".

Stan kubitu $|\psi\rangle$ jest opisywany za pomocą superpozycji stanów $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ i $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ jako wektor w postaci kolumnowej. W przypadku dwuwymiarowym wektor „ket" można zapisać jako:

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \alpha_0|0\rangle + \alpha_1|1\rangle, \tag{10.1}$$

gdzie $\alpha_0, \alpha_1 \in C$ są liczbami zespolonymi.

Wartości α_0, α_1 spełniają własność unormowania

$$|\alpha_0|^2 + |\alpha_1|^2 = 1. \tag{10.2}$$

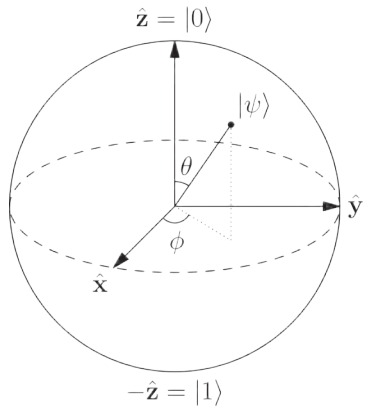
Zespolone współczynniki α_0, α_1 nazywamy **amplitudami**. Wykorzystując unormowanie 10.2 oraz fakt, iż istotne są tylko kwadraty amplitud (interpretowane jako prawdopodobieństwo), można przedstawić $|\psi\rangle$ jako:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \tag{10.3}$$

gdzie $\theta, \phi \in R$.

Pozwala to zaprezentować stan kubitu jako punkt na sferze Blocha.

Rysunek 10.2. Sfera Blocha – graficzna reprezentacja stanu pojedynczego kubitu



Źródło: opracowanie własne.

Wektory„bra" tworzone są jako sprzężenie hermitowskie i transpozycja wektorów ket.

$$\langle\psi| = \alpha_0^*\langle 0| + \alpha_1^*\langle 1| \;;\; \langle\psi| = |\psi\rangle^\dagger = [\alpha_0^* \;\; \alpha_1^*] \tag{10.4}$$

Notacja Diraca pozwala na „wygodny zapis" działań na wektorach stanu, takich jak **iloczyny skalarne** $\langle\psi|\phi\rangle$ lub **iloczyny tensorowe** służące do opisu systemów wielo-kubitowych.

Dla dwóch różnych kubitów: $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ oraz $|\phi\rangle = \beta_0|0\rangle + \beta_1|1\rangle$ iloczyn skalarny zdefiniowany jest jako:

$$\langle\psi|\phi\rangle = \alpha_0^*\beta_0\langle 0|0\rangle + \alpha_0^*\beta_1\langle 0|1\rangle + \alpha_1^*\beta_0\langle 1|0\rangle + \alpha_1^*\beta_1\langle 1|1\rangle = \alpha_0^*\beta_0 + \alpha_1^*\beta_1 \tag{10.5}$$

Iloczyn tensorowy wyraża się przez:

$$|\psi\rangle \otimes |\phi\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle, \tag{10.6}$$

gdzie $\sum_{i,j=0}^{1} |\alpha_i\beta_j|^2 = 1$.

W przypadku n klasycznych bitów istnieje 2^n możliwych różnych stanów klasycznych:

$$00..0, 00..1, \ldots, 1..11. \;\; (0, 1, \ldots, 2^n - 1) \tag{10.7}$$

Posiadając n kubitów, możemy utworzyć stan realizowany przez 2^n wektorów bazowych i tyle samo amplitud:

$$|00..0\rangle, |00..1\rangle, \ldots, |1..11\rangle, \ (|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle) \tag{10.8}$$

n kubitów opisuje stan naszego komputera kwantowego jako wektor jednostkowy, należący do przestrzeni generowanej iloczynem tensorowym (10.6) $\mathbb{C}^{2^n}$. Obliczenia realizowane są zazwyczaj od przygotowania stanu podstawowego, wyrażanego jako $|00..0\rangle = |0\rangle^{\otimes n}$. Dla uproszczenia zapisu tensorowego można ten stan określić jako $|0\rangle$.

Często można spotkać się z opinią, że **superpozycja** stanowi główną przewagę komputerów kwantowych nad klasycznymi. W rzeczywistości samo istnienie superpozycji nie wystarcza do rozwiązania wszystkich problemów obliczeniowych. Kluczowym wyzwaniem w praktycznym wykorzystaniu superpozycji jest proces pomiaru. W trakcie pomiaru stan kubitu nie jest bezpośrednio obserwowany w postaci superpozycji, lecz przechodzi w jeden ze stanów bazowych. Dla kubitu opisanego równaniem 10.8 wynikiem pomiaru może być stan $|0\rangle$ z prawdopodobieństwem $|\alpha_0|^2$ lub stan $|1\rangle$ z prawdopodobieństwem $|\alpha_1|^2$. W fizyce nazywamy ten efekt kolapsem wektora stanu.

W kontekście przetwarzania informacji mamy układ kwantowy, którego stan potrafimy przygotować i zmierzyć. Zostaje nam zatem opis operacji na kubitach. Do tego celu wprowadzimy pojęcie **operatora liniowego**. W ogólnym ujęciu operator A działa na stan i przekształca go w jakiś inny stan (z tej samej przestrzeni).

$$A|\psi\rangle = |\phi\rangle \tag{10.9}$$

W obliczeniach kwantowych będzie interesować nas szczególny rodzaj operatorów zdefiniowanych przez równanie własne:

$$A|\psi_a\rangle = a|\psi_a\rangle, \tag{10.10}$$

gdzie $|\psi_a\rangle$ nazywamy stanem własnym operatora A z wartością własną $a \in R$.

Operator A reprezentuje działanie bramki kwantowej. Ponieważ każdy kubit jest dwuwymiarowym wektorem, bramki (działające na n-kubitów) reprezentowane będą przez macierze unitarne $M_{2^n}(\mathbb{C})$ o wartościach zespolonych. Własność unitarności $U^\dagger U = UU^\dagger = I$ pozwala zachowywać prawdopodobieństwo (kwadrat normy wektora stanu). Unitarność operatorów w obliczeniach kwantowych wprowadza dodatkową, istotną cechę bramek – ich **odwracalność**. Przykładem nieparametryzowanych bramek działających na jeden kubit mogą być macierze Pauliego czy też bramka Hadamarda:

$$\sigma_x = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ \sigma_y = Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},$$
$$\sigma_z = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},\ H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{10.11}$$

Tak zdefiniowany operator można zapisać w postaci $A = a_i|\psi_i\rangle\langle\psi_i|$, gdzie a_i to wartości własne, natomiast $|\psi_i\rangle\langle\psi_i|$ to operatory rzutowe odpowiadające przestrzeni i-tego wektora własnego. Wynik pomiaru stanu komputera kwantowego odpowiada jednej z wartości własnej. Obliczając:

$$\langle\psi|A|\psi\rangle = \sum_{i,j}\langle\psi|\psi_i\rangle\langle\psi_i|\psi\rangle\langle\psi_i|A|\psi_i\rangle = \sum_i |\langle\psi|\psi_i\rangle|^2 a_i = \sum_i p(\psi_i)a_i \tag{10.12}$$

otrzymujemy wartość oczekiwaną operatora A w stanie $|\psi\rangle$, jeśli tylko $|\langle\psi|\psi_i\rangle|^2 = p(\psi_i)$ określimy jako prawdopodobieństwo, że po pomiarze nasz układ będzie w stanie własnym $|\psi_i\rangle$.

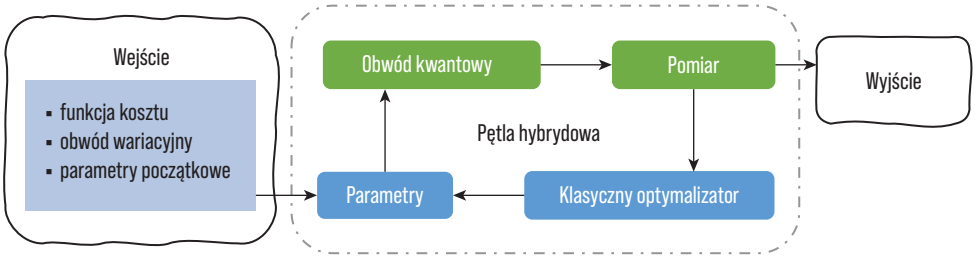
Posiadając zestaw bramek działających na kubity, możemy je wykorzystać do budowy **obwodów kwantowych**, stanowiących fundament realizacji algorytmów kwantowych. Te obwody składają się z ciągu operacji, które przekształcają stan początkowy kubitów za pomocą odwracalnych bramek kwantowych. Wykorzystanie takich obwodów może tworzyć przewagę nad klasycznymi algorytmami np. algorytm Deutsch-Josza [1992], Grovera [1996] czy Shora [1997].

Bramki mogą być zależne od parametrów, co pozwala nam tworzyć parametryzowane układy kwantowe. Przykładowe bramki przedstawia równanie 10.13.

$$R_X(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix},\ R_Y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix},$$
$$R_Z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{10.13}$$

Posiadając powyższe elementy, możemy zrealizować model obwodów kwantowych z wykorzystaniem PQC, zwanych też obwodami wariacyjnymi. Pozwala to realizować VQA w celu przybliżonego znajdowania minimum funkcji kosztu. Schemat wariacyjnego algorytmu kwantowego został przedstawiony na rysunku 10.3.

Rysunek 10.3. Schemat wariacyjnego algorytmu kwantowego



Źródło: opracowanie własne.

10.3. Kwantowe uczenie maszynowe a sztuczna inteligencja

Postęp technologiczny i powszechna cyfryzacja sprawiły, że dane stały się powszechnym i cennym zasobem [Zając, 2022]. Są one generowane i przetwarzane zarówno w sposób ustrukturyzowany, jak i nieustrukturyzowany. Strukturyzacja danych doprowadziła do rozwoju wielu modeli, które dziś ogólnie określamy jako modele uczenia maszynowego (*machine learning*, ML). Natomiast przetwarzanie danych nieustrukturyzowanych, takich jak tekst, obrazy czy wideo, przyczyniło się do rozwoju uczenia głębokiego (*deep learning*, DL). Oba te podejścia, często określane zbiorczo jako sztuczna inteligencja (*artificial inteligence*, AI), zostały stworzone głównie do rozpoznawania wzorców. Jednak coraz częściej wykorzystywane są również do modelowania i generowania nowych danych.

Klasyczny model sztucznej inteligencji możemy wyrazić jako funkcję $f(x;\theta)$, która zależy zarówno od danych reprezentowanych przez ustrukturyzowaną macierz x, jak i od parametrów θ, których wartości zostają ustalone w procesie uczenia.

W przypadku modeli kwantowych model uczenia maszynowego realizowany jest jako trzyczęściowy proces:

1. *Preprocessing* – pozwala załadować wektor danych $\vec{x}$ do parametrycznego obwodu kwantowego z wykorzystaniem tzw. *feature* mapy, czyli funkcji $\phi(\vec{x})$. Funkcja ta wyrażana jest jako parametryzowany obwód kwantowy, opisany przez operator $U_{\phi(\vec{x})}|0\rangle$. W tym miejscu warto zwrócić uwagę, iż procedura ta jest podobna do stosowania zanurzeń danych nieustrukturyzowanych w sieciach neuronowych (*embedding*), np. przetworzenie danych tekstowych z wykorzystaniem algorytmu Word2Vec. W fazie tej bardzo ważnym krokiem jest klasyczne przygotowanie danych poprzez takie elementy, jak czyszczenie braków danych, standaryzacja czy też transformacje pozwalające otrzymać odpowiedni rozkład zmiennej.

2. Parametryzowany kwantowy obwód wariacyjny – reprezentowany jest przez operator W_θ zależny od wektora parametrów θ, który działa na wektor stanu przygotowany przez obwód kodujący dane $W_\theta U_{\phi(\vec{x})}|0\rangle$. Ze względu na bardzo dużą ilość kombinacji bramek, które można łączyć równolegle i szeregowo, trudno jest wskazać jeden konkretny obwód pozwalający realizować wszystkie problemy analityczne (nie ma darmowych obiadów). Wybór odbywa się najczęściej poprzez ustalenie różnych schematów i ich trenowania [Sim, 2019]. Porównać można to do klasycznych sieci neuronowych, gdzie ilość warstw ukrytych, liczba neuronów w każdej warstwie czy funkcje aktywacji są tzw. hiper-parametrami, które ustala się przez doświadczenie. Wybrany schemat obwodów kwantowych określa się często mianem ansatzu.
3. Pomiar zrealizowanego obwodu – na tym etapie najczęściej estymuje się zbiór wartości oczekiwanych $\left\{\langle A_k\rangle_{x,\theta}\right\}_{k=1}^{K}$, przyjmujący wartości od −1 do 1. W zależności od realizowanego procesu można wykonać pomiar jednego lub większej ilości kubitów. Można również generować wynik w postaci amplitud, prawdopodobieństw czy też w postaci binarnej. Etap ten często nazywany jest postprocesingiem danych.

Do zakodowania informacji klasyczne komputery używają bitów (które mogą przyjąć wartość 0 lub 1). Aktualnie używa się systemów, które kodują znaki w postaci 32- lub 64-bitowych sekwencji. Pozwala to zakodować wszystkie znaki z kodowania ASCII bądź Unicode. Istnieje wiele różnych sposobów kodowania (*encode*) lub zanurzenia (*embed*) informacji w układ *n*-kubitów opisany przez stan kwantowy. Ponieważ chcemy użyć kwantowego komputera do uczenia się algorytmu na podstawie klasycznych danych, musimy zdecydować, w jaki sposób będziemy reprezentować wiersz danych, a nawet cały zbiór danych w postaci stanu kwantowego. W tradycyjnym ujęciu obliczeń kwantowych proces przygotowania komputera w stan początkowy nazywany jest przygotowaniem stanu (*state preparation*). W przypadku dużej ilości zmiennych można przeprowadzić redukcję wymiarów, wykorzystując np. mechanizm PCA lub inne metody opisane przez Przanowskiego [2021].

Jedną ze strategii jest tzw. kodowanie bazowe (*basis encoding*), które polega na zamianie wektora danych wyrażonego w postaci bitowej na wektory bazowe kubitów ($0 \to |0\rangle$ i $1 \to |1\rangle$). Chcąc jednak kodować dużo zmiennych i z dużą precyzją poszczególnych wartości, będziemy zmuszeni do wykorzystania bardzo dużej ilości kubitów, co dla obecnych maszyn nie jest zbyt dobrym rozwiązaniem. Kodowanie to, tak samo jak w przypadku bitów, pozwala realizować zarówno liczby całkowite, jak i rzeczywiste (dla z góry określonej precyzji).

Układ *n*-kubitów może być reprezentowany jako superpozycja stanów bazowych. Pozwala to zakodować dane w amplitudach (*amplitude encoding*). Normalizując wektor danych $x = \begin{pmatrix} x_1 \\ \vdots \\ x_{2^k} \end{pmatrix}$, tak by $\sum_k |x_k|^2 = 1$, możemy zakodować wszystkie x_k jako amplitudy. Na przykład $x = (0{,}073,\ -0{,}438,\ 0{,}730,\ 0.000) \leftrightarrow |x\rangle = 0{,}073|00\rangle - 0{,}438|01\rangle + 0{,}730|10\rangle + 0|11\rangle$

W niniejszej pracy zastosujemy inne kodowanie. Ponieważ wykorzystuje ono macierze rotacji $R_X(\theta)$, $R_Y(\theta)$, $R_Z(\theta)$ zdefiniowane w równaniu 10.13, kodowanie to nazywane jest kodowaniem rotacyjnym (*angle encoding*). Przetwarza ono zmienne na iloczyny i sumy funkcji cosinus i sinus. W kodowaniu tym istotne jest, aby podczas procesu skalowania zmiennych wyskalować je do wartości (−1,1).

$$|x\rangle = \bigotimes_{i=1}^{N} \cos(x_i)|0\rangle + \sin(x_i)|1\rangle$$

Informacje o pozostałych sposobach kodowania danych można znaleźć w artykułach Schuld, Bocharova, Svore i Wiebe'a [2021] oraz LaRose'a i Coyle'a [2020].

Analogicznie do uniwersalnego twierdzenia granicznego dla sieci neuronowych, zawsze istnieje obwód kwantowy, który może aproksymować funkcję z dowolnie małym błędem. W praktyce wiele wariacyjnych obwodów kwantowych realizowanych jest z ustaloną strukturą bramek, co – pomimo wykładniczego wzrostu ilości amplitud – pozwala na zredukowanie złożoności modeli poprzez małą liczbę wolnych do trenowania parametrów. Przy strategii ustalania szablonu bramek wykorzystywanych do uczenia modelu bierze się pod uwagę fakt, że dzisiejsze komputery kwantowe wciąć bazują na niedużej liczbie kubitów i można na nich realizować bardzo rzadko połączone bramki jedno- lub dwukubitowe. Najczęściej wykorzystuje się pojedyncze bramki obrotów jako elementy parametryzowane oraz bramki CNOT służące do generowania splątania pomiędzy poszczególnymi kubitami. Zarówno obwody wariacyjne, jak i sieci neuronowe mogą być traktowane jako warstwy połączonych składników kontrolowanych poprzez trenowalne parametry. Prowadzi to często do traktowania obwodów wariacyjnych jako kwantowych modeli sieci neuronowych. Obwody kwantowe realizują unitarne i jednocześnie liniowe bramki, podczas gdy istotną cechą sieci neuronowych jest zastosowanie nieliniowych funkcji aktywacji. Jednym ze sposobów realizacji nieliniowości w obwodach kwantowych jest wykorzystanie mechanizmu pomiaru kwantowego.

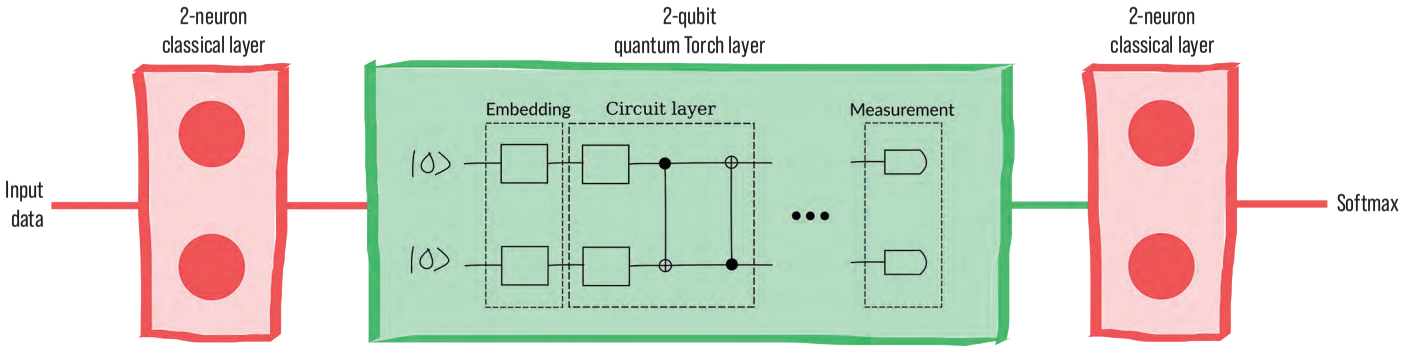
10.4. Hybrydowy algorytm kwantowej sieci neuronowej w procesie klasyfikacji binarnej

Poniżej przedstawiamy realizację procesu klasyfikacji binarnej z wykorzystaniem klasycznej sieci neuronowej oraz modelu hybrydowego, zawierającego klasyczne warstwy wejściowe i wyjściowe dla syntetycznie wygenerowanych danych. Dane wygenerowane zostały przy pomocy funkcji pochodzącej z pakietu scikit-learn *make_moons* (https://shorturl.at/Fjagn) z parametrami *n_sample = 100* oraz *noise = 0,4*. Sieci neuronowe zrealizowano z wykorzystaniem środowiska Python i biblioteki PyTorch. W roli klasycznego optymalizatora zastosowano funkcję Adam z parametrem uczenia 0,01, a jako funkcje straty – binarną entropię krzyżową (*binary cross entropy*). Wszystkie modele uczone były przez 100 epok. Pierwsza sieć neuronowa utworzona została z jedną warstwą liniową (dwie zmienne wejściowe i dwie wyjściowe) wraz z funkcją aktywacji softmax. Druga sieć neuronowa zawiera: warstwę wejściową zakończoną funkcją aktywacji Relu, warstwę ukrytą zakończoną kolejną funkcją aktywacji Relu oraz warstwę wyjściową zakończoną funkcją softmax. Trzecia sieć to hybrydowe połączenie klasycznej liniowej warstwy wejściowej z kwantowym obwodem dwu-kubitowym, zrealizowanym zgodnie ze schematem przedstawionym na rysunku 10.4. Schemat warstwy kwantowej pokazany został na rysunku 10.5.

Schemat czwartej sieci neuronowej, ukazującej możliwość składania warstw kwantowych w sposób równoległy, został zaprezentowany na rysunku 10.6.

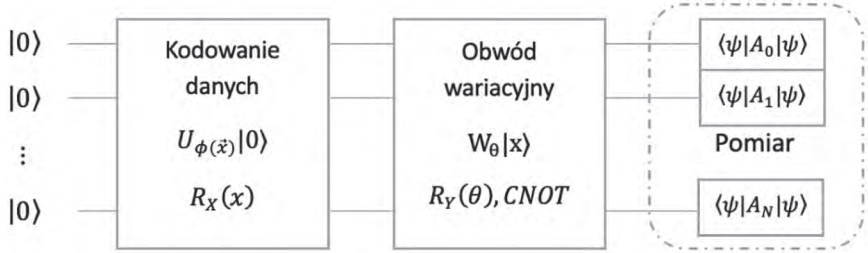
Jakość modeli przedstawiona została na podstawie metryki jakości dopasowania wyników (*accuracy*). Model klasycznej sieci bez warstw ukrytych (realizujący proces regresji logistycznej) osiągnął wynik 83% na zbiorze testowym. Sieć z warstwami ukrytymi wykazała typowy problem przeszacowania (*overfitting*), idealnie dopasowując się do danych treningowych. Jednak na zbiorze testowym osiągnęła zdolność tylko 73%. Pierwsza sieć kwantowa osiagnęła wynik 71% na zbiorze testowym i 84% na zbiorze treningowym. Sieć z warstwami kwantowymi złożonymi równolegle uzyskała 73% na zbiorze testowym i 91% na zbiorze treningowym.

Rysunek 10.4. Schemat hybrydowej kwantowej sieci neuronowej z klasyczną warstwą wejściową, warstwą kwantową zakończoną warstwą wyjściową z funkcją aktywacji softmax



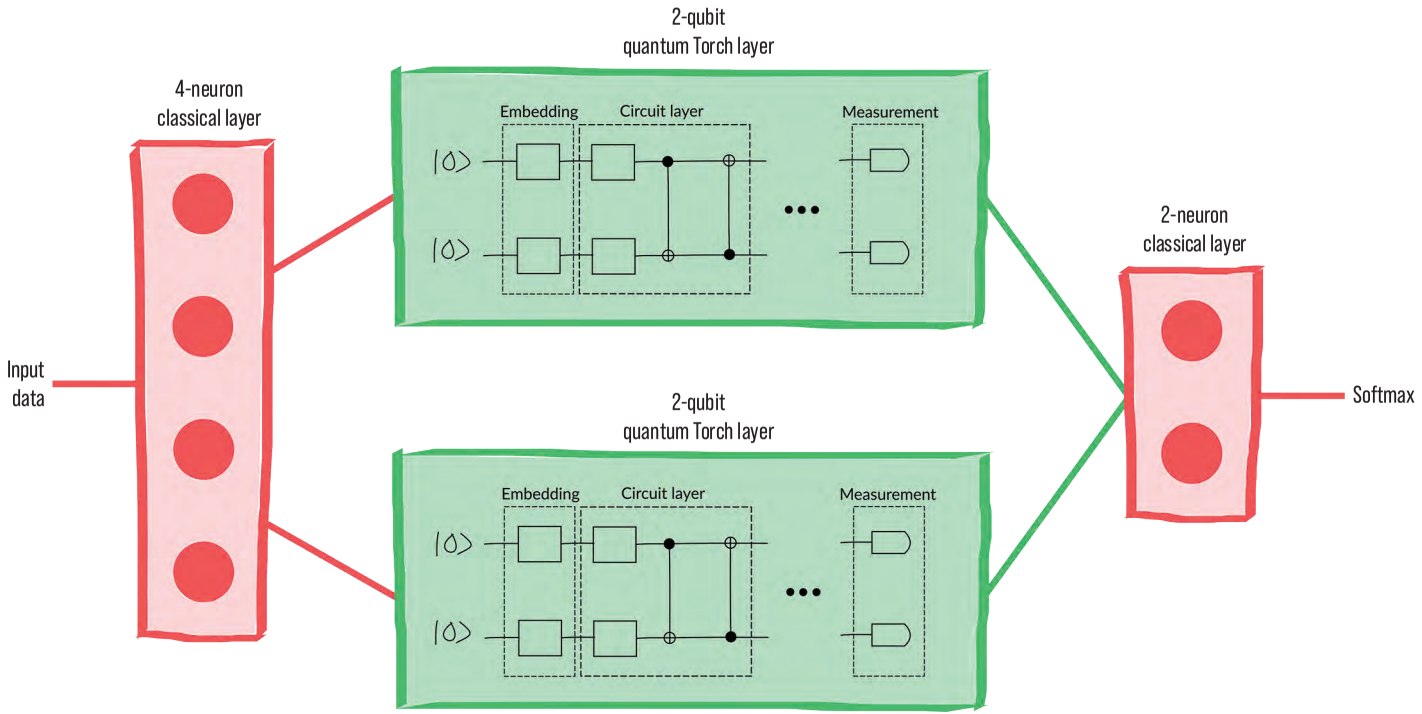
Źródło: Bromley [2024].

Rysunek 10.5. Schemat kwantowej warstwy sieci neuronowej z obwodem kodowania danych i obwodem wariacyjnym



Źródło: opracowanie własne.

Rysunek 10.6. **Schemat hybrydowej kwantowej sieci neuronowej z klasyczną warstwą wejściową, dwoma warstwami kwantowymi, zakończoną warstwą wyjściową z funkcją aktywacji softmax**



Źródło: Bromley [2024].

Podsumowanie

W niniejszej pracy przedstawione zostały podstawy matematyczne modeli kwantowego uczenia maszynowego oraz ich zastosowanie w problemie klasyfikacji binarnej. Przeanalizowano hybrydowe podejście bazujące na obwodach kwantowych i klasycznych optymalizatorach, realizujących warstwy ukryte klasycznej sieci neuronowej. Metody wariacyjnych algorytmów kwantowych mogą być wykorzystywane w problemach predykcyjnych oraz dla generowania nowych danych, a także w problemach optymalizacyjnych. Mimo że wciąż nie ma niezawodnego komputera kwantowego (duża liczba kubitów z długim czasem kontroli bramek oraz pełną korekcją błędów), pozwalającego w pełni stosować algorytmy kwantowe, metody hybrydowe mogą realizować w uproszczony (liniowy) sposób zadanie skomplikowanych sieci neuronowych.

Bibliografia

Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M. (2019). Parameterized Quantum Circuits as Machine Learning Models, *Quantum Science and Technology*, 4(4), 043001.

Bromley, T. (2024). *Turning Quantum Nodes Into Torch Layers*, https://pennylane.ai/qml/demos/tutorial_qnn_module_torch (dostęp: 7.10.2024).

Cerezo, M., Sone, A., Volkoff, T. Cincio, L., Coles, P.J. (2021). Cost Function Dependent Barren Plateaus in Shallow Parametrized Quantum Circuits, *Nature Communications*, 12, 1791.

Cybulski, J.L., Zając, S. (2024). Design Considerations for Denoising Quantum Time Series Autoencoder. W: *Computational Science – ICCS 2024. ICCS 2024. Lecture Notes in Computer Science* (vol. 14837), L. Franco, C. de Mulatier, M. Paszyński, V.V. Krzhizhanovskaya, J.J. Dongarra, P.M. A Sloot (Eds.). Cham: Springer.

Deutsch, D., Jozsa, R. (1992). Rapid Solution of Problems by Quantum Computation, *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907), s. 553–558.

Farhi, E., Goldstone, J., Gutmann, S. (2014.) *A Quantum Approximate Optimization Algorithm*. DOI: 10.48550/arXiv.1411.4028.

Feynman, R.P. (2022). *Lectures on Computation.* Boston: Addison-Wesley.

Gershenfeld, N.A., Chuang, I.L. (1996). Bulk Spin-Resonance Quantum Computation, *Science*, 275(5298), s. 350–356.

Grover, L.K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. W: *STOC '96: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (s. 212–219). New York: Association for Computing Machinery.

Harrow, A.W., Montanaro, A. (2017). Quantum Computational Supremacy, *Nature*, 549, s. 203–209.

Das, A., Chakrabarti, B. (2005). *Quantum Annealing and Related Optimization Methods*. Berlin: Springer-Verlag. DOI: 10.1007/11526216.

Havlíček, V., Mohseni, M., Lloyd, S. (2019). Supervised Learning with Quantum-Enhanced Feature Spaces, *Nature*, 567(7747), s. 209–212.

Hsu, C.-W., Ladd, T.D. (2020). Quantum Support Vector Classification, *Quantum*, 4, s. 328.

LaRose, R., Coyle, B. (2020). Robust Data Encodings for Quantum Classifiers, *Physical Review A*, 102(3), 032420.

Lund A.P., Bremner M.J., Ralph, T.C. (2017). Quantum Sampling Problems, Boson Sampling and Quantum Supremacy, *npj Quantum Information*, 3, 15.

Nayak, Ch., Simon, S.H., Stern, A., Freedman, M., Das Sarma, S. (2008). Non-Abelian Anyouns and Topological Quantum Computation, *Reviews of Modern Physics*, 80(3), s. 1083–1159.

Neumann von, J. (1945). *First Draft of a Report on the EDVAC*. Philadelphia: Moore School of Electrical Engineering, University of Pennsylvania.

Nielsen, M.A., Chuang, I. (2010). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M., Zhou, X., Love, P.J., Aspuru-Guzik, A., O'Brien, J.L. (2014). A Variational Eigenvalue Solver on a Photonic Quantum Processor, *Nature Communications*, 5(1), s. 4213.

Preskill, J. (2018). Quantum Computing in the NISQ Era and Beyond, *Quantum*, 2(79).

Przanowski, K, Zając, S. (2020). *Modelowanie dla biznesu. Metody machine learning, modele portfela consumer finance, modele rekurencyjne, analiza przeżycia, modele scoringowe*. Warszawa: Oficyna Wydawnicza SGH.

Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N. (2020). Circuit-Centric Quantum Classifiers, *Physical Review A*, 101(3), 032308.

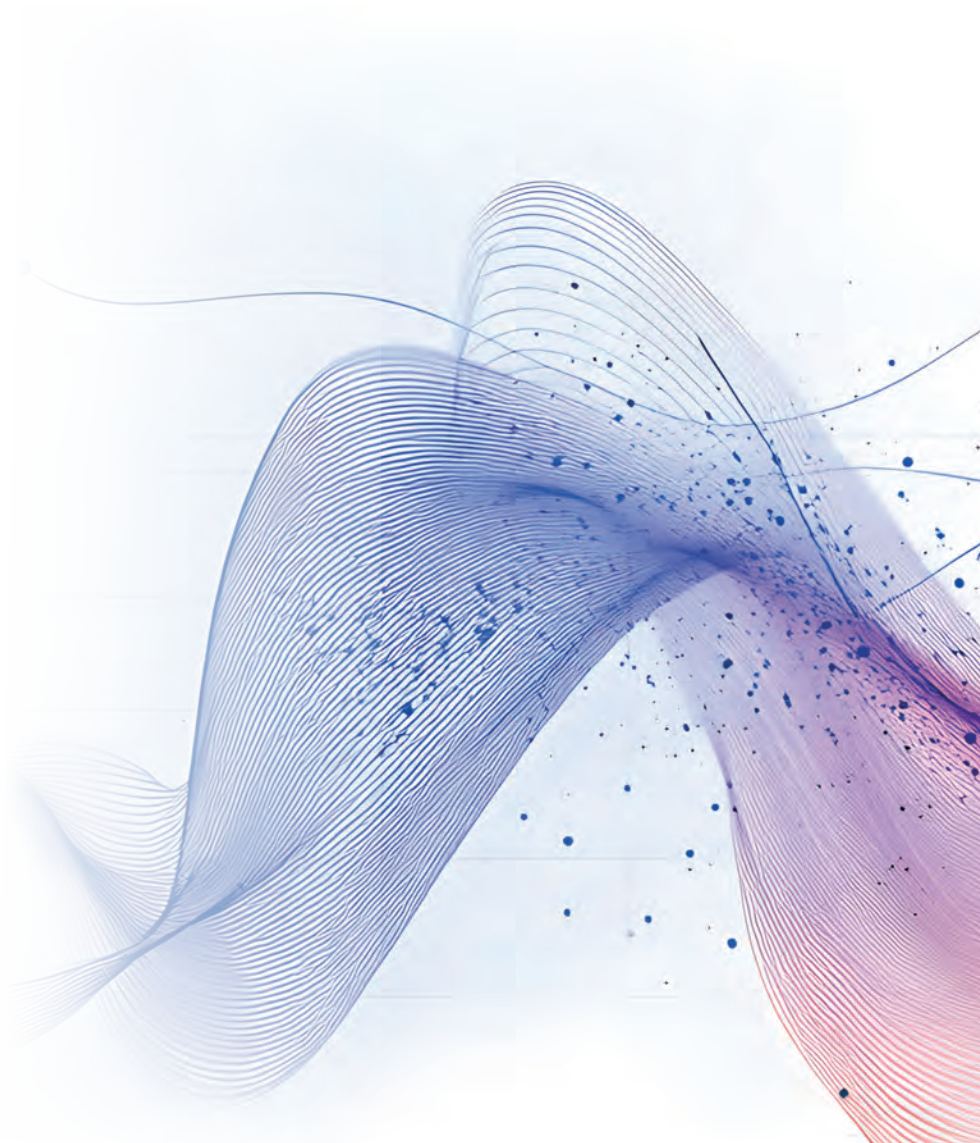
Shor, P.W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, *26*(5), s. 1484–1509.

Sim, S., Johnson, P.D., Aspuru-Guzik, A. (2019). Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms, *Advanced Quantum Technologies*, 2(12).

Turing, A. M. (1950). Computing Machinery and Intelligence, *Mind*, 59(236), s. 433–460.

Wong, T. G. (2022). *The Structure of Computer Systems*. Cambridge: Cambridge University Press.

Zając, S. (2022). *Modelowanie dla biznesu. Analityka w czasie rzeczywistym. Narzędzia informatyczne i biznesowe*. Warszawa: Oficyna Wydawnicza SGH.



OFICYNA WYDAWNICZA SGH
SZKOŁA GŁÓWNA HANDLOWA W WARSZAWIE
www.wydawnictwo.sgh.waw.pl

ISBN 978-83-8030-735-3