

МОДЕЛИ КВАНТОВОГО «СИЛЬНОГО» ВЫЧИСЛИТЕЛЬНОГО ИНТЕЛЛЕКТА И КВАНТОВЫХ НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ: ПЛАТФОРМА ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМИ РОБОТИЗИРОВАННЫМИ СОЦИОТЕХНИЧЕСКИМИ СИСТЕМАМИ

Капков Роман Юрьевич¹, Тятюшкина Ольга Юрьевна², Ульянов Сергей Викторович³

¹Преподаватель-исследователь;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: info@kapkov.pro.

²Кандидат технических наук, доцент;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: tyatyushkina@mail.ru.

³Доктор физико-математических наук, профессор;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

Главный научный сотрудник;

Объединенный институт ядерных исследований;

Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, 6;

e-mail: uyanovsv46_46@mail.ru.

Рассмотрены принципы и методы построения моделей квантового «сильного» вычислительного интеллекта на основе инструментария квантового глубокого обучения с применением квантовых нейронных сетей и квантового генетического алгоритма. Обсуждаются дополнительные особенности моделей квантового перцептрона и модели инженерии квантового глубокого обучения при применении инструментария оптимизации баз знаний интеллектуальных регуляторов на основе $QCOptKB^{TM}$ в задачах квантового интеллектуального управления роботизированными социотехническими системами в проектах «Индустрия 4.0 / 5.0 / 6.0». Описание особенностей квантового глубокого обучения позволяет точнее и глубже освоить возможности инструментария $QCOptKB^{TM}$, который включает в себя этапы обучения и извлечения (из исходных данных) сигнала обучения за счет инструментария $SCOptKB^{TM}$ на технологии мягких вычислений, и далее рассматривается как классические данные. За счет операторов квантовых вычислений классические данные кодируются кубитами, осуществляется оптимальный выбор квантовой корреляции между искомыми решениями и применением конструктивной интерференции извлекается искомый результат путем измерения. Таким образом, разработанный интеллектуальный инструментарий $SCOptKB^{TM}$ и $QCOptKB^{TM}$ включает принципы квантового глубокого обучения, и также как в случае технологии мягких вычислений, формирует оптимальную структуру теперь квантовой нейронной сети, а за счет применения квантового генетического алгоритма ускоряет поиск искомого решения.

Ключевые слова: роботизированные социотехнические производственные системы, Индустрия 5.0, квантовые сквозные ИТ, квантовая программная инженерия, квантовый «сильный» вычислительный интеллект, квантовые нейронные сети глубокого обучения.

Для цитирования:

Капков Р. Ю., Тятюшкина О. Ю., Ульянов С. В. Модели квантового «сильного» вычислительного интеллекта и квантовых нейронных сетей глубокого обучения: платформа интеллектуального управления промышленными роботизированными социотехническими системами // Системный анализ в науке и образовании: сетевое научное издание. 2024. № 3. С. 54-96. EDN: QANJQK. URL: <https://sanse.ru/index.php/sanse/article/view/627>.



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

MODELS OF QUANTUM «STRONG» COMPUTATIONAL INTELLIGENCE AND QUANTUM NEURAL NETWORKS OF DEEP LEARNING: PLATFORM FOR INTELLIGENT CONTROL OF INDUSTRIAL ROBOTIC SOCIOTECHNICAL SYSTEMS

Капков Роман Ю.¹, Тятюшкина Ольга Ю.², Улянов Сергей В.³

¹Teacher-researcher;

Dubna State University,

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: info@kapkov.pro.

²PhD in Engineering sciences, associate professor;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: tyatyushkina@mail.ru.

³Grand PhD in Physical and Mathematical Sciences, professor;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

Chief Researcher;

Joint Institute for Nuclear Research;

6 Joliot-Curie Str., Dubna, Moscow region, 141980, Russia;

e-mail: ulyanovsv46_46@mail.ru.

The principles and methods of quantum «strong» computational intelligence model's design based on the tools of quantum deep learning applying quantum neural networks and a quantum genetic algorithm are considered. Additional features of quantum perceptron models and quantum deep learning engineering models are discussed when using QCOptKB™ - knowledge base optimization toolkit for intelligent controllers in the tasks of quantum intelligent control of robotic sociotechnical systems in projects «Industry 4.0 / 5.0 / 6.0». The description of the features of quantum deep learning allows to more accurately and deeply master the capabilities of the QCOptKB™ toolkit, which includes the stages of learning and extracting (from source data) a learning signal using the SCOptKB™ toolkit based on soft computing technology, and is further considered as classical data. Due to quantum computing operators, classical data is encoded by qubits, the optimal choice of quantum correlation between the desired solutions and the use of constructive interference is made, the desired result is extracted by measurement. Thus, the developed intelligent SCOptKB™ and QCOptKB™ tools include the principles of quantum deep learning, and, as in the case of soft computing technology, forms the optimal structure of the new quantum neural network, and through the use of a quantum genetic algorithm accelerates the search for the desired solution.

Keywords: robotic sociotechnical production systems, Industry 5.0, quantum end-to-end IT, quantum software engineering, quantum «strong» computational intelligence, quantum neural networks of deep learning.

For citation:

Капков Р. Ю., Тятюшкина О. Ю., Улянов С. В. Модели квантового «сильного» вычислительного интеллекта и квантовых нейронных сетей на основе моделей квантового «сильного» вычислительного интеллекта. *System analysis in science and education*, 2024;(3):54-96 (in Russ). EDN: QAHJQK. Available from: <https://sanse.ru/index.php/sanse/article/view/627>.

Введение: Квантовое глубокое обучение на основе моделей квантовых нейронных сетей и квантового «сильного» вычислительного интеллекта

Около трех десятилетий назад квантовые вычисления представляли интерес в первую очередь для физиков-теоретиков и специалистов в области компьютерных технологий. Однако за последние несколько лет эта область достигла стремительного прогресса. В технологии квантовых вычислений быстро развиваются как теория, так и практика, и найдены интересные прикладные проблемно-ориен-

тированные области, в которых ИТ квантовых вычислений могут повлиять на качество решений, полученных с помощью классического машинного обучения. Преимущество квантовых компьютеров над классическими было продемонстрировано компанией *Google*, где было показано, что 53-кубитный квантовый компьютер выполняет случайные операции со схемой в течение нескольких секунд, на что современным суперкомпьютерам потребовались бы годы. *IBM* продемонстрировала надежное выполнение крупнейшей на сегодняшний день квантовой схемы, которая представляет собой «троттеризованную» схему, соответствующую модели Изинга с ближайшими соседями, со 127 кубитами и глубиной в два кубита, равной 60. За последние несколько лет *IBM* перешла на 433-кубитный процессор, объявив о запуске своего нового квантового устройства с 1121 кубитом в 2023 году.

Было лишь вопросом времени, когда пути квантовых вычислений пересекутся с другой областью, представляющей наибольший интерес и область применения в наше время: машинным обучением (*ML* – *machine learning*), где каждое из них извлекает выгоду из другого. Технология и методы *ML*, по сути, обучают компьютер извлекать информацию из данных без явного программирования. *ML* можно в широком смысле разделить на три категории: (i) контролируемое машинное обучение (обработка данных для прогнозирования результатов), такое как метод опорных векторов (*SVM* – *support vector machine*), нейронная сеть (*NN* – *neural networks*), наивный байесовский подход и т.д.; (ii) неуправляемое машинное обучение (работа с немаркированными данными), такое как *k*-средние алгоритм кластеризации, и (iii) обучение с подкреплением (обучение на основе окружающей среды и анализа ошибок агента).

Несколько популярных примеров применения классического *ML*: самоуправляемые роботы - автомобили, БПЛА, обнаружение мошенничества, обработка естественного языка, онлайн-рекомендации (от *Amazon*, *Walmart*, *Hulu*, *Netflix*).

Примечание. Первые работы по применению технологии квантовых вычислений в задачах интеллектуального управления [1-3], согласно [4], привели к возможности эффективного объединения идей машинного обучения и квантовых вычислений [5], а сама работа [5] признана как одна из самых цитируемых в рамках БД *SCOPUS* (см., Табл. В.1 из [4]).

Табл. 1. Пример десяти наиболее цитируемых работ по интеллектуальному управлению с применением квантовых вычислений

Title	Year	Citation
Multivariable system identification based on double quantum particle swarm optimization and big data Han and Yuan (2014a)	2014	23
Quantum Artificial Fish Swarm Algorithm Zhu and Jiang (2010)	2010	18
A review on application of particle swarm optimization in association rule mining Ankita et al. (2013)	2013	15
An Atomic-Array Optical Clock with Single-Atom Readout Madjarov et al. (2019)	2019	14
Quantum soft computing in control process design: Quantum genetic algorithms and quantum neural network approaches Ulyanov (2004)	2004	12
Relay selection scheme based on quantum differential evolution algorithm in relay networks Gao et al. (2017)	2017	9
An alert correlation method based on improved cluster algorithm Peng et al. (2008)	2008	9
The design of reversible gate and reversible sequential circuit based on DNA computing Song et al. (2008)	2008	9
Quantum-inspired swarm evolution algorithm Huang et al. (2007)	2007	9
Analog quantum computing (AQC) and the need for time-symmetric physics Werbos and Dolmatova (2016)	2016	8

Здесь кратко рассмотрим прикладные модели квантового машинного обучения, представим систематику подходов, а затем рассмотрим ключевые популярные в инженерии концепции параметризованной квантовой схемы (*PQC* – *parametrized quantum circuits*) и анзатц (*ansatz*). Предварительно опишем общую форму функций затрат, как функционал оптимизации в квантовом машинном обучении, кото-

рый зависит от ожидаемых значений наблюдаемых величин. Исходя из этого, в качестве предварительного шага обсудим вариационный квантовый собственный решатель (*VQE – variational quantum eigensolver*), который представлен как инструмент для решения определенного класса комбинаторных задач.

Модель VQE объединяется с методами квантового машинного обучения и основная идея использования классической оптимизации для минимизации ожидаемого значения наблюдаемой величины в качестве функции затрат. Рассмотрим вероятностные и детерминированные модели машинного обучения, полученные с помощью *PQCs*, с приложениями для неконтролируемого генеративного обучения, а также для контролируемого обучения в задачах регрессии и классификации.

Начатое обсуждение в [5], получило свое продолжение в виде дальнейшего обсуждения выбора *PQC* и проблемы кодирования данных в квантовых состояниях, а также некоторыми соображениями о будущих направлениях исследований.

Алгоритмы квантовых вычислений традиционно разрабатывались вручную без наложения ограничений на количество и надежность кубитов, квантовых вентилях и квантовых измерений. На практике современные квантовые компьютеры поддерживают работу с несколькими десятками кубитов, и, по прогнозам, в ближайшем будущем их число не превысит примерно 1000. Кроме того, подготовка квантового состояния, квантовые вентили и квантовые измерения в реальных квантовых компьютерах подвержены декогеренции и шуму. Это делает необходимым внедрение различных форм квантовой коррекции ошибок, чтобы имитировать бесшумную работу, предполагаемую формально разработанными квантовыми алгоритмами. А это, в свою очередь, еще больше увеличивает количество необходимых физических кубитов для обеспечения желаемого уровня надежности.

1. Методологии проектирования квантового машинного обучения

На этом фоне квантовое машинное обучение относится к альтернативной парадигме проектирования, показанной на рис. 1, основанной на интеграции маломасштабной квантовой схемы с классическим оптимизатором.

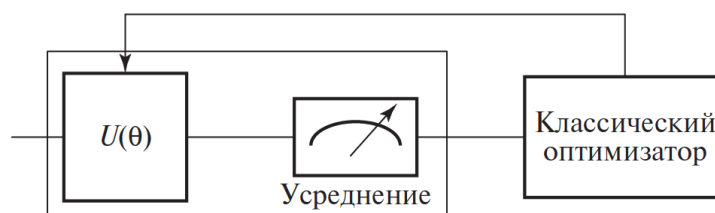


Рис. 1. Высокоуровневое описание методологии проектирования квантового машинного обучения

Как и классическое машинное обучение, квантовое машинное обучение использует индуктивную методологию, основанную на следующем двухэтапном подходе:

Выбор архитектуры параметризованной квантовой схемы (PQC) (также известной как *ansatz – анзац*). Разработчик сначала выбирает архитектуру *PQC*, задавая последовательность параметризованных квантовых элементов. Параметризованные квантовые вентили могут, например, включать в себя матрицы вращения Паули, углы которых рассматриваются как свободные параметры, и которые могут быть оптимизированы. В целом, работа *PQC* определяется унитарной матрицей $U(\theta)$, которая зависит от вектора свободных параметров θ . Выбор *PQC* в идеале должен быть продиктован имеющимися у разработчика знаниями о квантовых алгоритмических архитектурах, которые хорошо подходят для решения поставленной задачи. Этап выбора архитектуры *PQC* отражает проблему выбора архитектуры нейронной сети в классическом машинном обучении.

Параметрическая оптимизация. *PQC*, реализующий унитарное значение $U(\theta)$, подключен к классическому оптимизатору. Оптимизатор загружается измерениями квантового состояния, производимые *PQC*, обычно в форме предполагаемых ожиданий наблюдаемых величин; и это приводит к обновлению вектора параметров θ . Обновления направлены на минимизацию некоторой функции затрат, которая может зависеть также от данных обучения. Этот этап параметрической оптимизации является

общим для классической методологии машинного обучения, в которой параметры модели θ оптимизируются с помощью методов, основанных на градиентном спуске, с целью минимизации потерь при обучении.

Параметризованная квантовая схема (PQC), реализующая унитарную матрицу $U(\theta)$, оптимизируется с помощью вектора параметров θ на основе измерений выходных данных PQC.

Система квантового машинного обучения, изображенная на рис. 1, обладает рядом потенциальных преимуществ по сравнению со стандартным подходом ручной разработки квантовых алгоритмов, предполагающим отказоустойчивость квантовых компьютеров:

- Поддерживая квантовый компьютер в рабочем состоянии, классический оптимизатор может напрямую учитывать не идеальности и ограничения доступных квантовых ресурсов;
- Если в PQC оператор $U(\theta)$ достаточно гибок, а классический оптимизатор достаточно эффективен, этот подход может автоматически создавать хорошо работающие квантовые алгоритмы, которые было бы трудно создать «вручную» с помощью формальной методологии проектирования.

3. Таксономия квантового машинного обучения

В отличие от классического машинного обучения, в котором данные и обработка являются классическими, в квантовом машинном обучении данные и/или обработка являются квантовыми. Четыре возможных сценария обобщены в табл. 2 и рассмотрены в этом разделе.

Табл. 2. Систематизация настроек машинного обучения в зависимости от типа данных и обработки (С - классический, Q - квантовый)

		Данные		Обработка	
		С	Q	С	Q
Данные	С	СС	СQ		
	Q	QC	QQ		

В случае классических данных и классической обработки (СС) предусмотрена оптимизация стандартной модели машинного обучения, например, нейронной сети, с помощью классического оптимизатора. Как показано на рис. 2 (а), классическая модель машинного обучения реализует параметризованную функцию $h(x|\theta)$, преобразующую классические входные данные x в выходные данные.

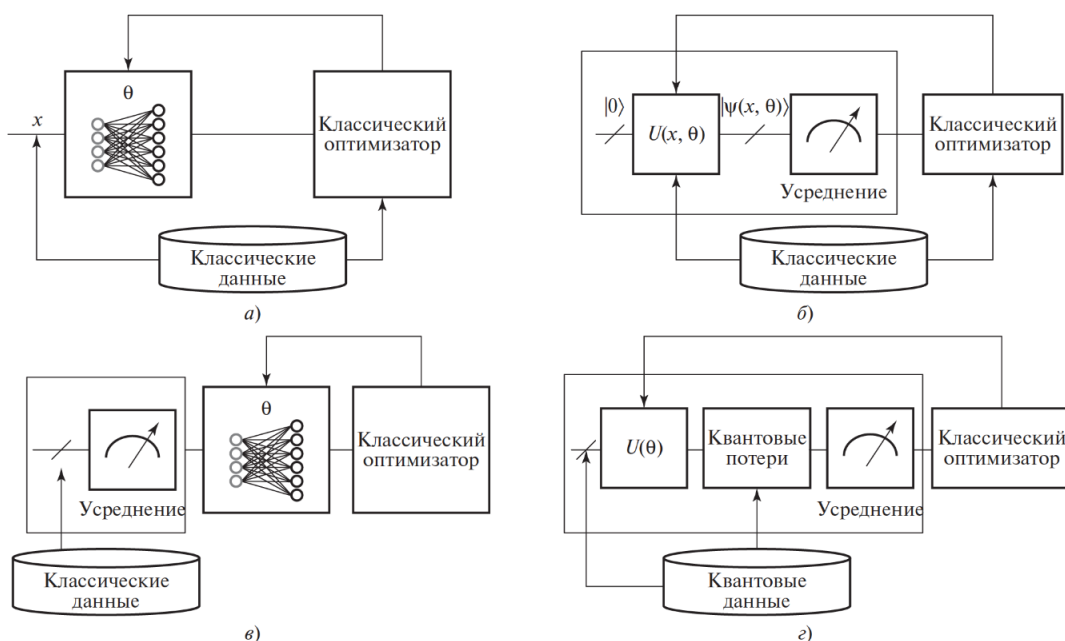


Рис. 2. Виды машинного обучения

Классический оптимизатор минимизирует функцию потерь, зависящую от классических данных, по вектору θ . Например, при контролируемом обучении классические данные состоят из пар входных данных x и соответствующих желаемых выходных данных y . Более того, функция потерь определяется потерями при обучении, которые измеряют, насколько хорошо предсказания выходных данных, полученные с помощью функции $h(x|\theta)$, соответствуют желаемым выходным данным y в доступных данных.

В настоящее время наиболее распространенная система квантового машинного обучения характеризуется классическими данными и квантовой обработкой (CQ). Если взять в качестве примера случай контролируемого обучения, как показано на рис. 2 (б), то квантовая модель обычно определяется PQC, которая реализует унитарное преобразование $U(x, \theta)$. Унитарное значение $U(x, \theta)$ зависит как от классического входного значения x , так и от вектора параметров модели θ . Входные данные для PQC представляют собой набор кубитов в основном состоянии $|0\rangle$, а выходное состояние, создаваемое PQC, задается формулой $|\psi(x, \theta)\rangle = U(x, \theta)|0\rangle$. Выходное состояние $|\psi(x, \theta)\rangle$ также известно как квантовое встраивание классических данных x , поскольку оно преобразует классические входные данные x в квантовое состояние. Выходные данные измерений, возможно, после усреднения, кодируют предсказание модели целевой переменной y .

Классический оптимизатор минимизирует некоторую функцию потерь по вектору параметров модели θ , который измеряет, насколько хорошо результат измерения квантового состояния $|\psi(x, \theta)\rangle$ соответствует целевому значению y . Функция потерь выражается в терминах математического ожидания наблюдаемой величины для квантового вложения $|\psi(x, \theta)\rangle$. Для оценки математического ожидания данной наблюдаемой величины, определяющей ее стоимость, требуется выполнить применение PQC несколько раз, чтобы получить эмпирическое среднее значение.

Как показано на рис. 2 (в), при использовании квантовых данных и классической обработки (QC) обучающийся получает доступ к квантовым данным. Квантовые данные состоят из набора квантовых систем, состояние которых описывается матрицей плотности. Квантовые данные могут быть получены квантовым датчиком и сохранены в квантовой памяти – при этом как квантовое зондирование, так и квантовое хранение являются двумя активными областями исследований. В случае контроля качества сначала измеряются квантовые данные, а затем результаты классических измерений обрабатываются классической моделью машинного обучения. Машинное обучение контролю качества может быть применено к квантовой томографии, которая представляет собой задачу определения свойств квантового состояния на основе результатов измерений.

В случае квантовых данных и квантовой обработки (QQ), как показано на рис. 2 (г), квантовые данные обрабатываются PQC. Функция потерь оценивается в квантовом компьютере путем сравнения, используя подходящую квантовую схему, выходных данных PQC с целевыми квантовыми данными. Примером машинного обучения QQ являются квантовые генерирующие состязательные сети, в которых квантовые потери вычисляются с помощью детектора, который пытается отличить реальные квантовые данные от квантовых состояний, генерируемых PQC. Другой пример — квантовые вариационные автоэнкодеры, которые нацелены на сжатие квантового состояния в меньший набор кубитов.

Сосредоточимся исключительно на примере CQ, который лучше изучен и потенциально более подходит для инженерных применений. Тем не менее, обсуждение PQC в следующем разделе также актуально для решений QQ.

4. Анзатц и параметризованные квантовые схемы

Как обсуждалось в предыдущем разделе, в основе методов квантового машинного обучения лежат PQC, которые являются темой этого раздела.

PQC реализует унитарное преобразование $U(\theta)$, действие которого на n кубитах зависит от настраиваемого (классического) вектора параметров θ , состоящего из d действительных чисел. Как было показано в предыдущем разделе, в случае CQ унитарность также может зависеть от входных данных x , и обсудим эту ситуацию в контексте контролируемого обучения ниже. Унитарный $U(\theta)$ обычно задается

как последовательность одно- или двух-кубитных вентилях, причем каждый вентиль, возможно, зависит от параметра θ .

Примечание. Выбор архитектуры PQC в виде последовательности определенных элементов подобен выбору класса модели в классическом машинном обучении, например, в архитектуре нейронной сети. В квантовом машинном обучении называют архитектуру PQC анзатц (*ansatz*). Этот термин происходит от немецкого слова, означающего «подход» или «попытка». Что касается класса моделей в машинном обучении, то следует выбирать *ansatz*, по возможности, исходя из знаний предметной области. Например, в задачах квантовой химии существуют известные алгоритмические структуры, которые, как известно, хорошо отражают природу определенных явлений.

В оставшейся части этого раздела рассмотрим типичные ответы, расположив их в порядке возрастания сложности и, следовательно, выразительности результирующего PQC . В частности, сначала рассмотрим параметризованные одно-кубитные вентиля и аналог среднего поля; затем опишем более сложные аппаратно-эффективные аналоги; и, наконец, обсудим параметризованные много-кубитные вентиля.

Пример: Параметризованные однокубитные гейты. Наиболее распространенный тип параметризованных квантовых вентилях представлен одно-кубитными матрицами вращения Паули. Матрица P -вращения Паули $R_P(\theta)$ с $P \in \{X, Y, Z\}$ является унитарной матрицей, генератор которой задается

через $\frac{\theta}{2} \cdot P$, т.е.,

$$R_P(\theta) = \exp\left(-i\frac{\theta}{2}P\right) = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)P,$$

где θ - действительный параметр. Параметр θ также известен как угол поворота гейта $R_P(\theta)$. Три варианта поворота по Паули приведены в 3.

Табл. 3. Наиболее часто используемые параметризованные гейты Паули с однократным вращением

Вращение Паули	Определение
$R_X(\theta)$	$\exp\left(-i\frac{\theta}{2}X\right) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$
$R_Y(\theta)$	$\exp\left(-i\frac{\theta}{2}Y\right) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$
$R_Z(\theta)$	$\exp\left(-i\frac{\theta}{2}Z\right) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) - i\sin\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) + i\sin\left(\frac{\theta}{2}\right) \end{bmatrix}$

Имеем равенство $R_X(-\pi) = iX$, $R_Y(-\pi) = iY$, и $R_Z(-\pi) = iZ$. Таким образом, помимо общей фазы, вращения Паули восстанавливаются в виде частных случаев, т.е. при определенном выборе угла поворота θ задаются значения X , Y и Z .

Общий параметризованный одно-кубитный гейт может быть выражен, помимо глобальной фазы, в виде следующего каскада двух типов вращений Паули:

$$R(\theta^1, \theta^2, \theta^3) = R_P(\theta^1)R_{P'}(\theta^2)R_P(\theta^3), \text{ где } P' \neq P, \text{ и } P, P' \in \{X, Y, Z\}.$$

Типичный вариант - $P = Z$ и $P' = Y$. Общий одно-кубитный вентиль (6.6) определяется тремя параметрами, или углами поворота, θ^1, θ^2 и θ^3 .

Одно-кубитные гейты могут быть применены параллельно к кубитам квантовой системы. Чтобы указать кубит $k \in \{0, 1, \dots, n-1\}$, к которому применяется одно-кубитный элемент управления, обычно добавляют индекс k к обозначению элемента управления. В качестве примера, обозначение Z_k описывает унитарный элемент, который применяет Z -элемент Паули только к k -му кубиту. Это преобразование сводится к строке Паули, содержащей все единичные матрицы I размерности 2×2 , за исключением позиции k , где имеем оператор Паули Z , т.е., $Z_k = \underbrace{I \otimes \dots \otimes I}_{k-1} \otimes Z \otimes \underbrace{I \otimes \dots \otimes I}_{n-k}$. Аналогично, можем написать:

$$R_k(\theta_k^1, \theta_k^2, \theta_k^3) = \underbrace{I \otimes \dots \otimes I}_{k-1} \otimes R(\theta_k^1, \theta_k^2, \theta_k^3) \otimes \underbrace{I \otimes \dots \otimes I}_{n-k}$$

для обычного одно-кубитного вентиля с параметрами $(\theta_k^1, \theta_k^2, \theta_k^3)$, применяемыми к k -му кубиту.

5. Анзатц усредненного поля

Анзатц $U(\theta)$, использующий только однокубитные гейты, известен как «анзатц усредненного поля». Как показано на рис. 3, анзатц среднего поля определяет унитарные элементы вида

$$U(\theta) = R(\theta_0^1, \theta_0^2, \theta_0^3) \otimes \dots \otimes R(\theta_{n-1}^1, \theta_{n-1}^2, \theta_{n-1}^3) = \prod_{k=1}^{n-1} R_k(\theta_k^1, \theta_k^2, \theta_k^3),$$

где $(\theta_k^1, \theta_k^2, \theta_k^3)$ для $k \in \{0, 1, \dots, n-1\}$ являются скалярными параметрами. В целом, вектор параметров $\theta = [\theta_0^1, \theta_0^2, \theta_0^3, \dots, \theta_{n-1}^1, \theta_{n-1}^2, \theta_{n-1}^3]^T$ содержит $d = 3n$ параметров.

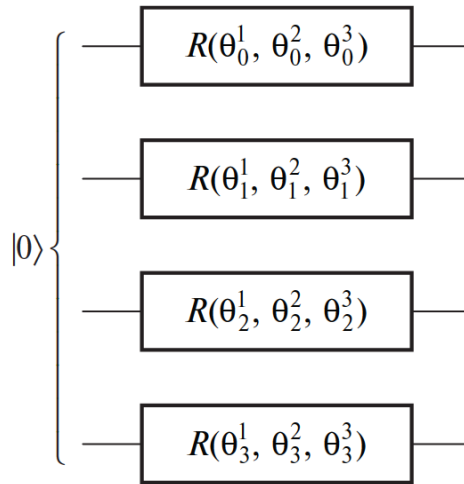


Рис. 3. Анализатор усредненного поля состоит из обычных одно-кубитных вентилях, применяемых параллельно к n кубитам, которые изначально находятся в основном состоянии $|0\rangle$ (на рисунке $n = 4$)

Учитывая входное состояние $|0\rangle$, QCS, следующий за анализатором среднего поля, выводит разделяемое состояние $U(\theta)|0\rangle = R(\theta_0^1, \theta_0^2, \theta_0^3)|0\rangle \otimes \dots \otimes R(\theta_{n-1}^1, \theta_{n-1}^2, \theta_{n-1}^3)|0\rangle$, где записали $|0\rangle$ как многокубитное основное состояние, так и соответствующие однокубитные состояния с некоторыми переобозначениями.

Фактически, QCS, следующий за анализатором среднего поля, может привести к получению всех возможных векторов в вычислительной базе. Этот результат может быть получен, установив

$\theta_k^1 = 0$ и $\theta_k^3 = 0$ для всех кубитов $k \in \{0, 1, \dots, n-1\}$ и путем выбора параметров θ_k^2 в виде $\theta_k^2 = 0$ или $\theta_k^2 = \pi$ для кодирования двух вычислительных базисных векторов.

Однако, в более общем плане, решение многих практических задач может быть полезным благодаря анзатцам, которые могут создавать запутанность между кубитами. Это может быть сделано путем введения запутывающих многокубитных вентилях в PQC, что позволяет выйти за рамки анализа среднего поля.

6. Аппаратно-эффективный анзатц

Переходя на один уровень выше по оси сложности для анзатца, аппаратно-эффективный анзатц включает в себя как параметризованные однокубитные гейты, так и фиксированный запутывающий унитарный оператор U_{ent} , который не зависит от вектора параметров θ . Как подробно описано ниже, запутывающий унитарный оператор U_{ent} обычно состоит из гейтов в виде элементов CNOT или CZ. Для простоты реализации используется фиксированный запутывающий унитарный оператор, поскольку двухкубитные элементы управления, как правило, трудно реализовать на квантовых компьютерах. Эта мотивация оправдывает название «аппаратно-эффективный» для этого анзатца.

В частности, как показано на рис. 4, аппаратно-эффективный анзатц предписывает PQC, которые реализуют каскад из L уровней унитарных операторов в виде

$$U(\theta) = U_L(\theta) \cdot U_{L-1}(\theta) \cdots U_1(\theta),$$

где каждая унитарная матрица $U_l(\theta)$ на l -м уровне может быть выражена как

$$U_L(\theta) = U_{ent} \left(R(\theta_{l,0}^1, \theta_{l,0}^2, \theta_{l,0}^3) \otimes \cdots \otimes R(\theta_{l,n-1}^1, \theta_{l,n-1}^2, \theta_{l,n-1}^3) \right).$$

с помощью $(\theta_{l,k}^1, \theta_{l,k}^2, \theta_{l,k}^3)$ для $k \in \{0, 1, \dots, n-1\}$.

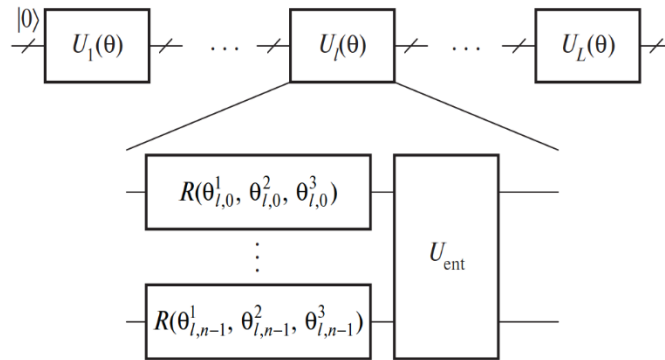


Рис. 4. Аппаратно-эффективный аналог состоит из L уровней, каждый из которых включает общие однокубитные вентили, подключенные параллельно к n кубитам, и фиксированную квантовую схему оператора запутывающих состояний U_{ent} .

Кубиты изначально находятся в основном состоянии $|0\rangle$. Следовательно, вектор параметров θ включает в себя $d = 3nL$ параметров, причем $3n$ параметров описывают каждый слой.

Таким образом, PQC, следующий за аппаратно-зависимым анзатцом, применяет на каждом уровне l отдельные однокубитные вентили ко всем n кубитам, как в анзатц со средним полем, за которым следует фиксированный запутывающий унитарный оператор U_{ent} . Запутывающий унитарный оператор U_{ent} обычно реализуется с использованием линейных преобразований, гейтов вращения или полных квантовых схем запутывания:

- Линейные схемы запутывания реализуют двух-кубитные логические элементы между последовательными кубитами. В примере на рис. 5 используются логические элементы CZ , и это может быть выражено как

$$U_{ent} = \prod_{k=0}^{n-2} C_{k,k+1}^Z,$$

где $C_{k,k+1}^Z$ – гейт CZ между кубитами k и $k + 1$.

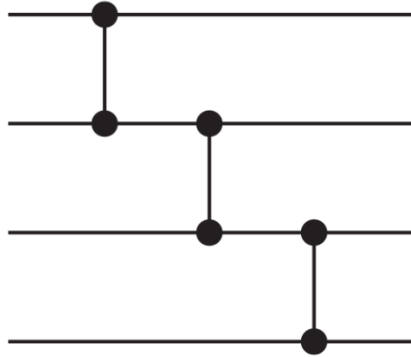


Рис. 5. Линейная схема запутывания для $n = 4$ кубитов, реализованная с помощью CZ -вентилей

- Гейты квантовых схем запутанных состояний добавляют к архитектуре линейных схем запутывания в виде дополнительного двухконтурного элемента между первым и последним кубитами.
- Схемы полного запутывания реализуют двух-кубитные элементы управления между всеми парами кубитов, как показано на рис. 6 с элементами CZ .

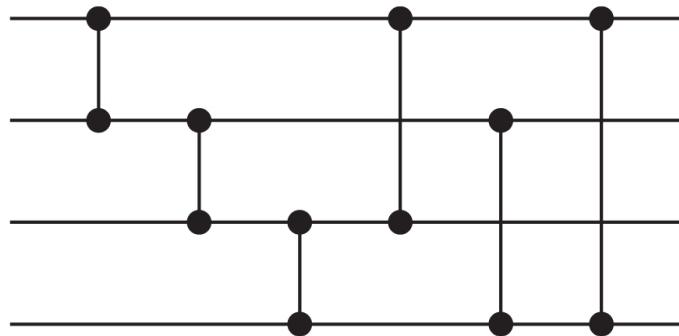


Рис. 6. Полная схема запутывания для $n = 4$ кубитов, реализованная с помощью CZ -вентилей

В связи с этим отметим, что элементы CZ имеют преимущество перед элементами $CNOT$ в том, что их можно применять в любом порядке, поскольку они коммутируют.

Аппаратно-эффективный анзац является универсальным, и часто считается, что он играет ту же роль, что и полностью подключенные классические нейронные сети при разработке квантового машинного обучения. Тем не менее, важно подчеркнуть, что аппаратно-эффективный аналог не обладает свойствами, аналогичными свойствам полностью подключенных классических нейронных сетей, когда речь заходит о зависимости от параметров модели.

Примечательно, что в полностью связанных классических нейронных сетях имеется значительная свобода в оптимизации связей между нейронами путем определения индивидуальных синаптических весов. Напротив, в аппаратно-эффективном анзаце можно управлять только вращениями, применяемыми к каждому отдельному кубиту, а зависимости между кубитами определяются фиксированными квантовыми схемами запутывания.

7. Сравнение классического и квантового обучения: модели классического и квантового перцептрона

Перцептрон обучается, проверяя, совпадает ли выходное значение y с истинным или ожидаемым значением. Если такое сопоставление не произошло на основе какой-либо заранее определенной метрики, то параметры нейрона (w, b) оптимизируются таким образом, чтобы выходные данные сети соответствовали желаемому значению. В перцептроне Розенблатта эта оптимизация была выполнена путем простого добавления входного вектора x к весам w , если перцептрон недооценивал выходное значение по сравнению с истинной меткой, и вычитания x , если перцептрон переоценивал выходное значение по сравнению с истинной меткой. Смещение b также было изменено на ± 1 в обоих случаях соответственно. Как только выходные данные перцептрона совпадают с меткой, считается, что нейрон «научился» выполнять задачу.

Каждый из описанных перцептронов, по сути, эквивалентен биологическому нейрону. Нейронная сеть с прямой связью получается путем наложения множества таких нейронов слой за слоем таким образом, что нейроны в одном слое соединяются с нейронами в другом слое. В рабочем режиме сеть моделирует нелинейную функцию вида $f: x \rightarrow y$, где $x \in \mathbb{R}^d$ и $y \in \mathbb{R}^m$; d -мерный вектор x является входным, а m -мерный вектор y - выходным. Если в сети есть L слоев, состоящих из нейронов, это означает, что в первом (входном) и последнем (выходном) слоях будет d и m нейронов соответственно. Промежуточные слои называются скрытыми слоями.

Сосредоточим внимание только на k -м и $(k-1)$ -м слоях. Аффинное преобразование, определенное на k -м уровне, будет параметризовано весовой матрицей $W \in \mathbb{R}^{p \times q}$, где q - количество нейронов в k -м слое, а p - количество нейронов в $(k-1)$ -м слое, а также вектором смещения $b^k \in \mathbb{R}^q$. Преобразование воздействует на реакцию активации $(k-1)$ -го уровня, т.е. $\sigma(z^{k-1}) \in \mathbb{R}^p$, следующим образом:

$$z^k = W\sigma(z^{k-1}) + b^k.$$

Таким образом, в результате преобразования получается новый вектор $z^k \in \mathbb{R}^q$, который проходит через процесс активации с использованием любой из функций активации, определенных ранее, и передается в следующий слой. Этот процесс повторяется до тех пор, пока не будет достигнут последний слой. В конце на L -м уровне ответом на активацию является $z^L = y$. Это значение сравнивается с истинными значениями/метками данных (например, y^*). Можно определить множество таких показателей для уравнения, одним из простых примеров является норма L^2 вектора разности $\|y - y^*\|_2$ или перекрестная энтропия (cross-entropy), если y и y^* являются распределениями вероятностей и т.д.

Такие показатели часто называют функциями полезности или затратами. Как только функция затрат определена, определяется ошибка в метрике, и это используется для оценки градиента функции затрат относительно параметров смещения L -го слоя и весовых коэффициентов взаимосвязи между L -м и $(L-1)$ -м слоями. Затем процесс повторяется для всех слоев, пока не будет достигнут первый слой. В конце концов, пользователь получает доступ к градиенту функции стоимости в зависимости от настраиваемых параметров всех слоев. Этот метод получения градиента называется обратным распространением.

Схема структуры квантово-классической гибридной нейронной сети для реализации ANN приведена [6] на рис. 7 (а).

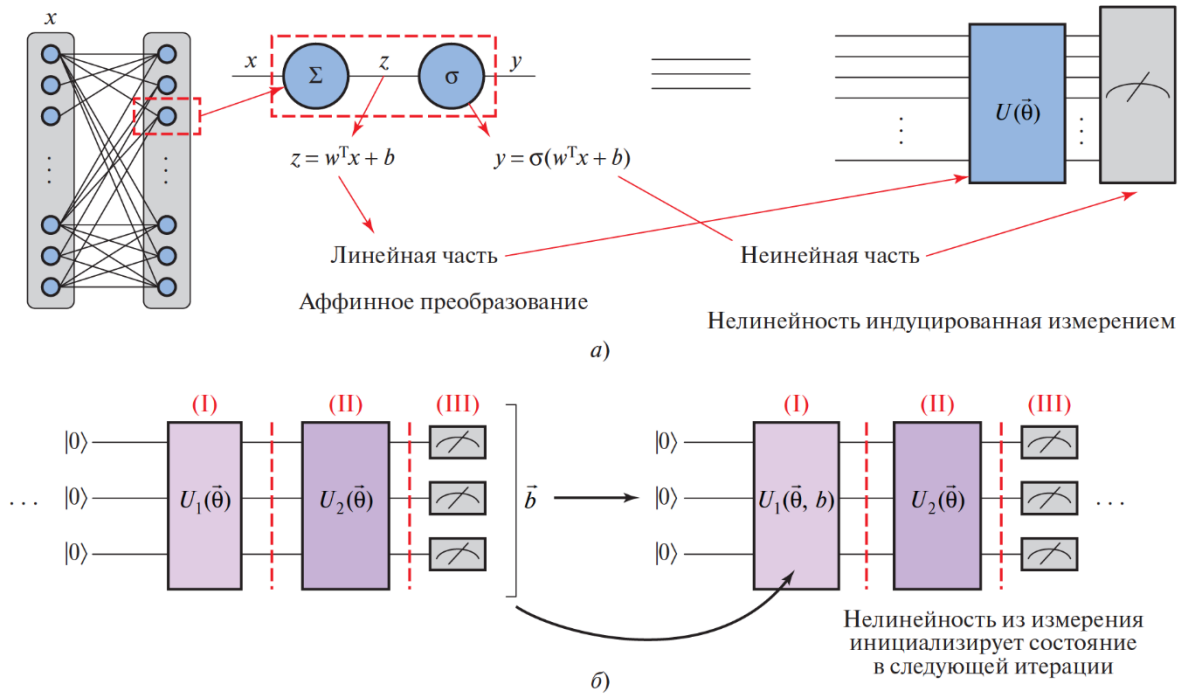


Рис. 7. (а) Схема структуры квантово-классической гибридной нейронной сети для реализации ANN; (б) Пример построения гибридной квантово-классической нейронной сети для 3-х кубитов

Линейная часть выполняется с помощью параметризованных унитарных преобразований, которые определяют настраиваемые веса и смещения сети, тогда как нелинейная активация получается из измерений в квантово-классической гибридной нейронной сети.

Этап (I) на рис. 7 (б) относится к кодированию состояния с единичным значением U_1 , этап (II) - фактическая вариационная схема с параметризованным единичным значением U_2 , а этап (III) - измерение для воспроизведения эффекта нелинейной активации. На следующей итерации результаты измерений \vec{b} представлены для применения в унитарном U_1 для кодирования состояния. Таким образом, нейронная сеть работает с полной обратной связью. Обучение осуществляется путем изменения параметров U_2 .

Классический слой в квантово-классической гибридной нейронной сети должен выполнять функцию активации, соединяющую различные квантовые слои. На рис. 7 (а) представлена схема квантово-классических гибридных нейронных сетей, где линейная часть в классической нейронной сети заменена квантовыми цепями.

На рис. 8 (б) показан пример построения гибридной квантово-классической нейронной сети для 3-х кубитов (см., рис. 7).

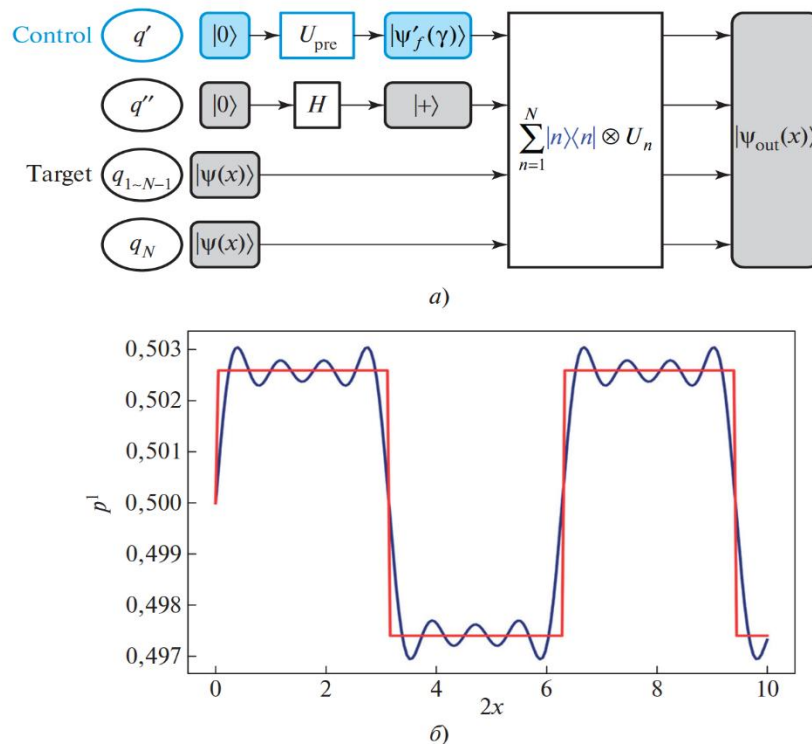


Рис. 8. (а) основная структура квантовых схем, оценивающих произвольные периодические функции. (Существует два основных модуля. Первый содержит U_{pre} , действующий на вспомогательные кубиты q' , и гейты Адамара, действующие на q'' . Следующий успешно воспроизводимый модуль формируется из N контролируемых унитарных операций, обозначаемых как U_n . q' (синий цвет) - управляющие кубиты. q' преобразуются в состояние $|\psi'_{f'}(\gamma)\rangle$ в соответствии с операцией U_{pre} , где γ определяется как F_N на рис. 8 (б); синяя кривая представляет конечный результат выхода квантовой схемы, оценивающей прямоугольные волновые функции. В то же время красная кривая представляет собой исходную форму прямоугольных волновых функций

На рис. 8 (а) представлена схема структуры схемы, а на рис. 8 (б) показана аппроксимация периодических прямоугольных волновых функций. Квантовая схема содержит N кубитов для хранения информации о различных N компонентах Фурье и $M + 2$ вспомогательных кубита с $M = \lceil \log_2 N \rceil$ для операций управления. Требуемый выходной сигнал будет измерен в последнем кубите q_N с временной сложностью вычисления $O(N^2 \lceil \log_2 N \rceil^2)$, что при определенных обстоятельствах приводит к полиномиальному ускорению. В заключение, важным, но все еще открытым вопросом является поиск оптимального подхода к реализации нелинейных функций активации в квантовых нейронных сетях.

8. Особенности квантового глубокого обучения

Машинное обучение и квантовые вычисления развивались параллельно. В данном введении кратко коснемся обсудим применение квантовых вычислений для поиска более мощных и эффективных моделей машинного обучения. Квантовое машинное обучение (*QML – quantum machine learning*) - современная теоретическая область, которая находится на стыке квантовых вычислений (*QC – quantum computing*) и машинного обучения (*ML – machine learning*). В классическом машинном обучении для моделирования задачи цель состоит в том, чтобы найти функцию f , заданную x и y , такую, что: $y = f(x)$. Предоставляем входные данные и ожидаемый результат (метку) *ML*-модели, которая изучает правила, не сообщая компьютеру явно, как решить проблему. Программа учится делать это сама. Функция потерь (математическое выражение, которое вычисляет величину, на которую алгоритм не достиг точной цели) проверяет, насколько правильным является решение машинного обучения. Во время обуче-

ния модели ML становится очевидным, что в начале не все из них верны. QML также стремится минимизировать функцию потерь, используя свойство, называемое квантовым туннелированием (QT – *quantum tunneling*). QT выполняет полный поиск в пространстве функций потерь, чтобы найти значение с минимальными потерями. Существует несколько популярных методов, которые QC использует для решения задач ML , таких как квантовая нейронная сеть (QNN – *Quantum Neural Networks*), квантовый анализ главных компонент ($QPCA$ – *Quantum Principal Component Analysis*), квантовые методы опорных векторов ($QSVM$ – *Quantum Support Vector machines*), квантовое обучение с подкреплением (QRL – *Quantum Reinforcement Learning*), квантовая оптимизация (QO – *Quantum Optimization*) и т.д.

Алгоритм QML может работать с классическими или квантовыми данными. Когда алгоритм QML работает с классическими данными, возникает необходимость кодировать данные в кубиты, которые передаются в QML . Поэтому предварительно обсудим несколько широко используемых методов кодирования, которые используются для кодирования информации из классических данных в кубиты. Затем рассмотрим некоторые алгоритмы, такие как QNN , $QSVM$ и др.

А. Кодирование классических данных в кубиты

Квантовая система не может напрямую считывать классические данные. Поэтому необходимо кодировать информацию, содержащуюся в данных, в кубиты. Эту идею кодирования не следует путать с кодированием при квантовой коррекции ошибок. В данном случае кодирование просто подразумевает загрузку классических данных в кубиты. Хотя не существует четкого правила относительно природы кодирования, наиболее широко используются следующие три метода.

1. *Базовое кодирование*: Классические данные представлены в виде последовательности битов (аналогично «горячему» кодированию в классическом машинном обучении). Например, если $x \in \{0, 1\}^n$ – «горячее» кодирование конкретных классических данных, соответствующее квантовое кодирование равно $|x\rangle$. Для двух различных $|x\rangle \neq |x'\rangle$ имеем $\langle x|x'\rangle = 0$. Допустим, что (x_1, \dots, x_m) – m классических входных данных для обучения. Тогда соответствующие квантовые данные имеют вид:

$$|\psi\rangle_{in} = \frac{1}{\sqrt{m}} \sum_{i=1}^m |x_i\rangle.$$

Этот метод является простейшей формой кодирования классических данных в квантовом устройстве. Тем не менее, он часто требует значительного количества кубитов, особенно если требуется кодировать дроби, и, таким образом, не позволяет использовать меньшее количество кубитов.

2. *Амплитудное кодирование*: значение классических данных кодируется как амплитуда вероятности базовых состояний. Например, если классические данные представлены в виде $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ $\alpha_i \in \mathbb{R} \forall i$, то соответствующий квантовый ввод имеет вид:

$$|\psi\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^m \frac{\alpha_i}{|\alpha_i|} |x_i\rangle.$$

3. *Угловое кодирование*: значения объектов также могут быть закодированы в углы амплитуды кубитов. Рассмотрим набор данных с $2n$ характеристиками объектов. Этот метод требует только n кубит для кодирования $2n$ характеристик объектов, и кубиты могут быть реализованы с использованием схем малой глубины. Значения x_i и x_{i+1} для объектов i и $i+1$ соответственно могут быть закодированы в один кубит следующим образом:

$$|\psi\rangle_{i,i+1} = \cos(x_i)|0\rangle + e^{i x_i+1} \sin(x_i)|1\rangle.$$

До сих пор ни одна система кодирования не доказала, что она превосходит другие в QML . Исследователи использовали один или несколько из этих методов кодирования в зависимости от конкретной задачи. После кодирования данных в кубиты алгоритм QML пытается обучаться на этих данных, чтобы получить желаемые результаты. Далее обсудим два широких класса алгоритмов QML , а именно квантовую нейронную сеть (*Quantum Neural Network* – QNN) и квантовый метод опорных векторов (*Quantum Support Vector machine* – QVM).

Б. Классическое и квантовое обучение: качественное сравнение моделей

Классические и квантовые модели машинного обучения (*ML – machine learning*) проиллюстрированы на рис. 9.

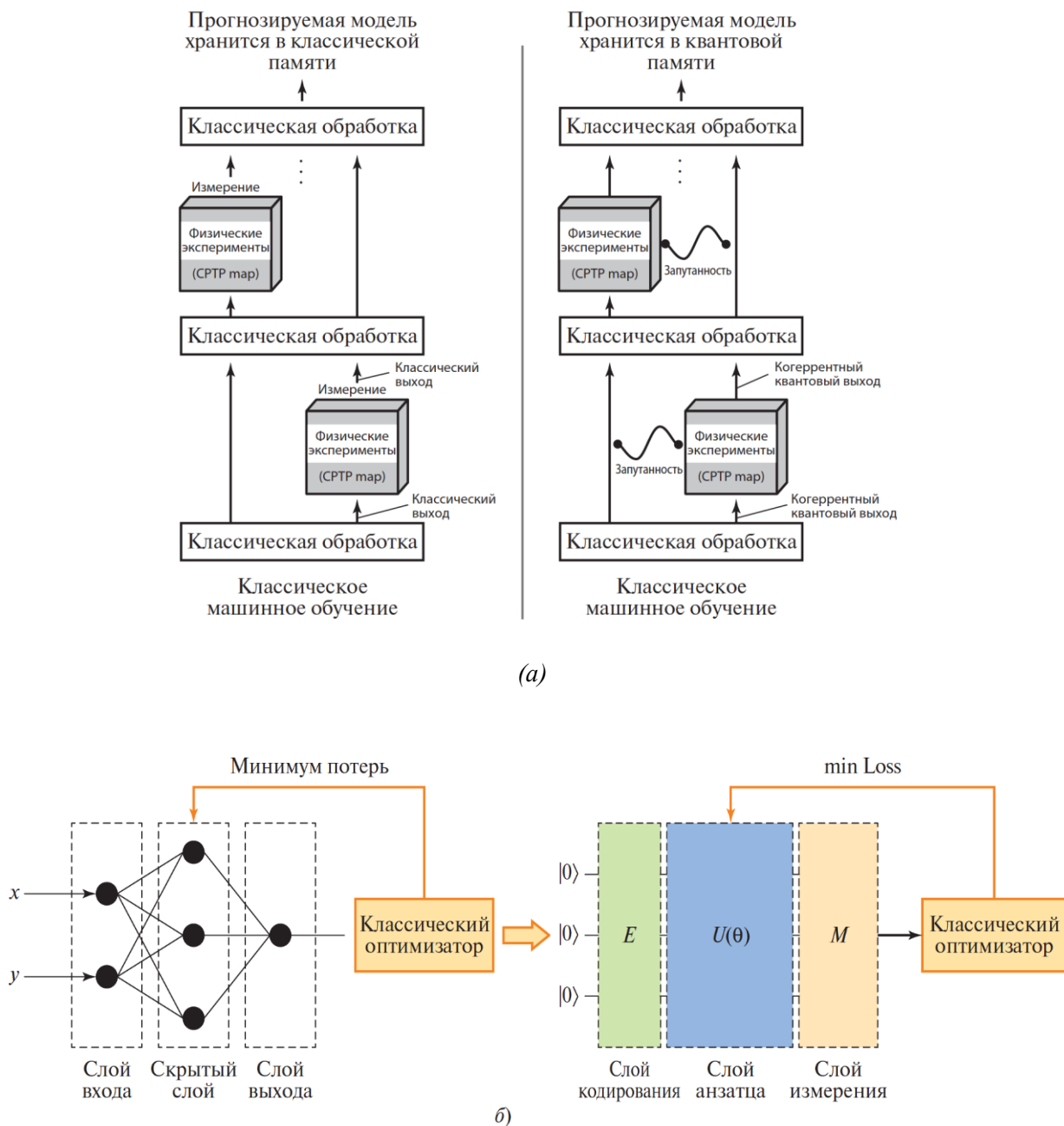


Рис.9. Сравнение классической и квантовой моделей машинного обучения (а); Классическая нейронная сеть и структурная схема квантовой нейронной сети [6,7]

Согласно классическим нейронным сетям, каждый уровень квантовых нейронных сетей может быть сопоставлен с классическими нейронными сетями.

Общая структура *QNN* показана на рис. 9 (б). По сравнению с искусственными нейронными сетями уровень кодирования данных в *QNN* эквивалентен входному уровню, кодирующему классические данные в количественные состояния, а конкретный метод кодирования выбирается в соответствии с реальной ситуацией. Уровень квантового преобразования эквивалентен скрытому слою. Уровень квантового преобразования использует *PQC*, разработанный с использованием параметрических квантовых элементов, для преобразования и обработки входного квантового состояния, которое включает в себя вес параметрического квантового элемента, эквивалентного традиционной нейронной сети.

Уровень квантовых измерений эквивалентен уровню вывода, который может получать измеренное значение конечного квантового состояния. Среди них классический оптимизатор используется для оптимизации и обновления параметров квантового вентиля, чтобы получить модель, соответствующую требованиям. Исследования показывают, что по сравнению с традиционными нейронными сетями, *QNN*, основанная на *PQC*, более выразительна с точки зрения выбранных параметров [8,9].

Цель состоит в том, чтобы определить вид неизвестного полностью положительно определенного отображения с сохранением следов (*CPTP* – *completely positive and trace preserving*) \mathcal{E} путем проведения физических экспериментов. На этапе изучения классической модели после каждого запроса \mathcal{E} к ней выполняется измерение; результаты классических измерений, собранные на этапе изучения, используются на этапе прогнозирования. На этапе обучения квантовой модели несколько запросов к \mathcal{E} могут быть включены в единую когерентную квантовую схему, что приводит к получению выходного состояния, сохраненного в квантовой памяти; к этому сохраненному квантовому состоянию обращаются на этапе прогнозирования [6-9].

Предварительно отметим, что *QNN* – подкласс вариационных квантовых алгоритмов, который состоит из квантовых схем, содержащих параметризованные логические операции. Принципиальная схема, сравнивающая классическую и квантовую *NN*, приведена на рис. 10, и они могут быть сопоставимы, если имеют одинаковый размер входных и выходных данных и количество обучаемых параметров.

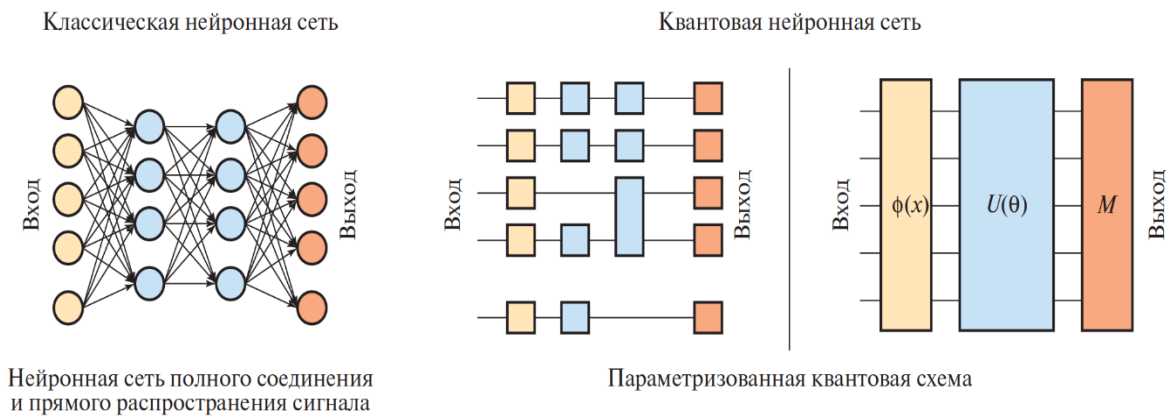


Рис. 10. Классическая нейронная сеть против квантовой нейронной сети [7]

На рис. 11 представлена иллюстрация сходства между классическими и квантовыми нейронными сетями.

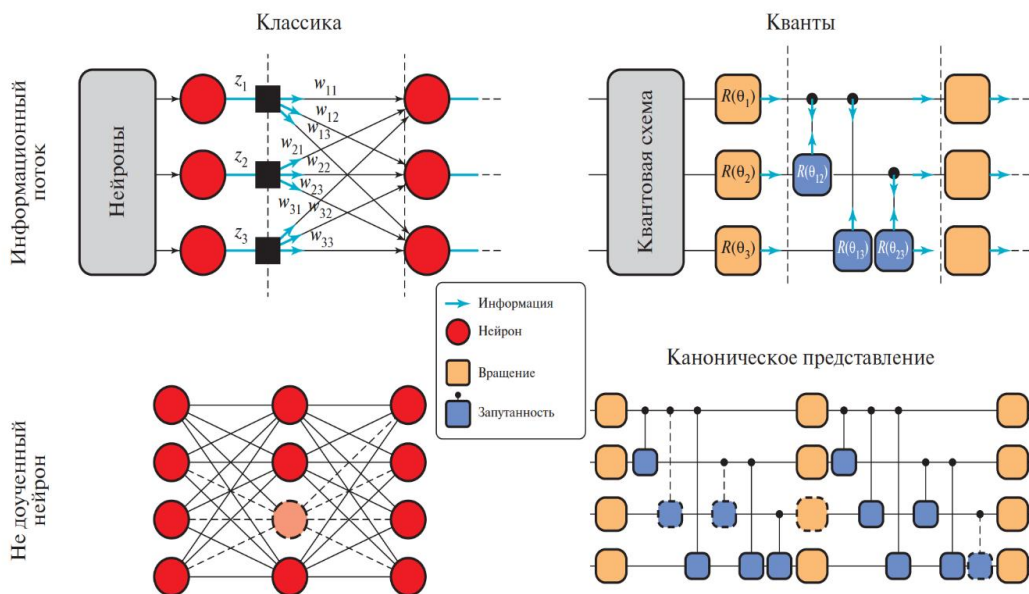


Рис. 11. Примеры подобия между классическими и квантовыми нейронными сетями [10]

На рис. 12 (а) показана нейронная сеть, которая имеет входной слой с m узлами, один скрытый слой с l узлами и выходной слой с n узлами.

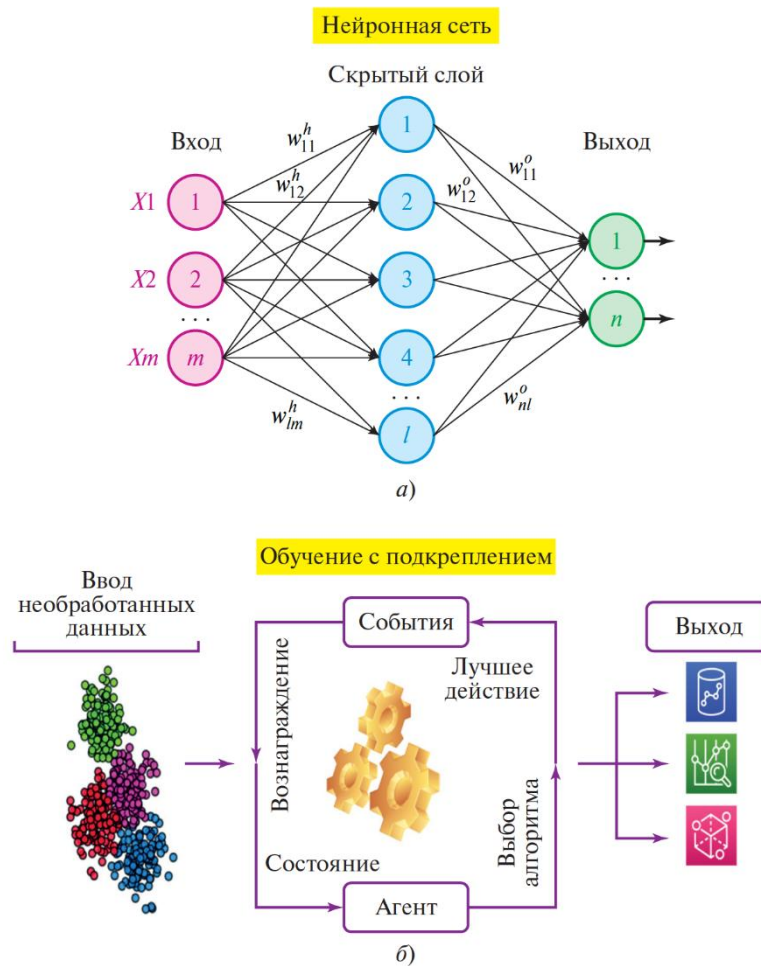


Рис. 12. Принципиальная схема (а) искусственной нейронной сети [Источник изображения: <https://medium.com/swlh/neural-networks-4b6f719f9d75>] и (б) обучения с подкреплением [Источник изображения: <https://www.inwinstack.com/blog-en/blog-ai-en/6262/>]

Обучение с подкреплением (*RL – reinforcement learning*) – область машинного обучения, которая обучает агента выбирать действие из пространства действий (где среда фиксирована), чтобы максимизировать вознаграждение за определенное время (см., рис.12 (б)). Обучение с подкреплением не контролируется и не неконтролируемо. Агент получает награду или наказание в зависимости от выбранного действия. Алгоритм учится выбирать те действия, которые максимизируют его вознаграждение. В RL есть четыре важных элемента:

- Агент – программа, которая обучена выполнять определенную работу;
- Окружающая среда – реальный или виртуальный мир, в котором находится агент;
- Действие – движение, которое совершает агент, вызывая изменение статуса в окружающей среде;
- Вознаграждение – оценка после действия, совершенного агентом, и это может быть как положительным, так и отрицательным фактором.

При построении оптимальной стратегии агент может столкнуться с дилеммой между исследованием новых состояний и одновременной максимизацией общего вознаграждения. Например, агент не может знать, получил ли он уже достаточно хорошее вознаграждение, и дальнейшее изучение новых состояний просто снизит его ценность. Это называется исследованием или эксплуатацией. Наилучшая

общая стратегия предполагает краткосрочные потери для принятия наилучшего общего решения. Обучение с подкреплением находит применение во многих областях, где обучающие данные об обучении могут быть недоступны.

Примечание. В классическом машинном обучении «переобучение» - явление, возникающее, когда данная модель чрезмерно хорошо усваивает обучающие данные и, таким образом, плохо работает с «невидимыми» или маскирующими данными. Широко используемым методом машинного обучения является так называемый «отсев», который предотвращает чрезмерную специализацию вычислительных блоков и, следовательно, снижает риск переобучения. С появлением квантовых нейронных сетей в качестве моделей обучения переобучение вскоре может стать проблемой из-за увеличения глубины квантовых схем, а также многократного внедрения классических функций, которые используются для задания вычислительной нелинейности. Здесь представляем обобщенный подход к применению метода отсева в моделях квантовых нейронных сетей, определяя и анализируя различные стратегии квантового отсева, чтобы избежать переобучения и достичь высокого уровня обобщения. Исследование позволяет представить силу квантового отсева в обеспечении возможности обобщения, предоставляя полезные рекомендации по определению максимального снижения переобучения.

В. Оптимизация отображения характеристик в квантовых схемах

Первые квантовые схемы состояли всего из десятков вентилях и до 5 кубитов. Недавно были проведены эксперименты со схемами, имеющими 127 кубитов и глубину вентиля 60, с одиночными кубитными вентилями в диапазоне тысяч и двух-кубитными вентилями в диапазоне сотен, а также схемами с 433 кубитами. Число возможных начальных сопоставлений для первой схемы составит $127!$, а последующие этапы планирования и легализации требуют огромного пространства для поиска.

Существует разрыв между квантовыми ресурсами, необходимыми для выполнения современных квантовых алгоритмов, и ресурсами, доступными в современных зашумленных квантовых устройствах промежуточного уровня (*NISQ*), поскольку процессоры состоят из слабо взаимосвязанного отображения взаимодействия, что ограничивает взаимодействия между кубитами.

На рис. 13 (а) приведен пример схемы подключения 5-кубитного процессора *Quito IBM Q*.

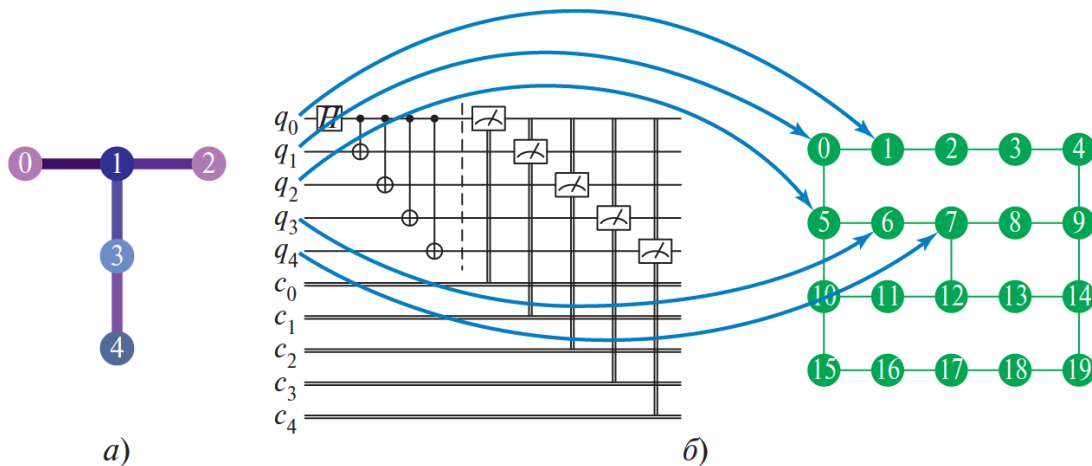


Рис. 13. (а) Схема сопряжения 5-кубитного IBM-квантового процессора *Quito*, (б) графическое представление схемы сопряжения, где 5-кубитная схема сопоставляется с 20-кубитным квантовым процессором с заданным отображением сопряжения

Это отображение связей определяет набор $(0,1), (1,0), (1,2), (2,1), (1,3), (3,1), (3,4), (4,3)$, в котором кортежи являются целевыми и управляющими соответственно в возможной операции *CNOT*. Таким образом, восемь из двадцати пар (т.е. $n^2 - n$) могут использоваться для работы *CNOT* в цепи. Это ограничивает возможности, предоставляемые квантовыми устройствами. Следовательно, для минимизации использования элементов подкачки для выполнения операций *CNOT* между другими парами кубитов необходимы эффективные методы отображения схем.

Был предложен метод построения схемы, основанный на алгоритме, который предназначен для поиска наилучших мест размещения путем сравнения с желаемым результатом. Общим недостатком

этого метода является то, что желаемый результат часто неизвестен. Оптимальность метода компоновки может быть выражена многими способами, например, путем оптимизации глубины расположения цепей. *QCL* использует входную схему C_{in} для преобразования ее в функционально эквивалентную схему C_{out} . Глубина выходной цепи может быть больше, чем у входной цепи, и соотношение между глубиной выходной и входной цепи является критерием эффективности методов *QCL*. Минимизация этого соотношения известна как оптимизация *QCL*. Представленный метод компоновки *QXX*, включает настраиваемую гауссову функцию для оценки глубины генерируемых цепей и определения области схемы, которая влияет на большую часть глубины. Параметры модели *QXX* оптимизированы с использованием импровизированного взвешенного случайного поиска.

Хотя промежуточного уровня (*NISQ*) еще не способны выполнять крупномасштабные вычисления общего назначения, исследователи изучили гибридные квантово-классические алгоритмы, которые могут быть выполнены в этих устройствах. Эти алгоритмы, как правило, разделены на модули, некоторые из которых переданы на аутсорсинг классическим процессорам (*CPU*). Таким образом, квантовая схема имеет небольшую глубину и менее восприимчива к шуму. Другими словами, эти алгоритмы все еще могут выдавать приемлемые результаты в условиях шума. Эти алгоритмы находят применение в квантовой химии, комбинаторной оптимизации, квантовом машинном обучении и т.д.

На рис. 14 приведена схема работы квантово-классических гибридных алгоритмов.

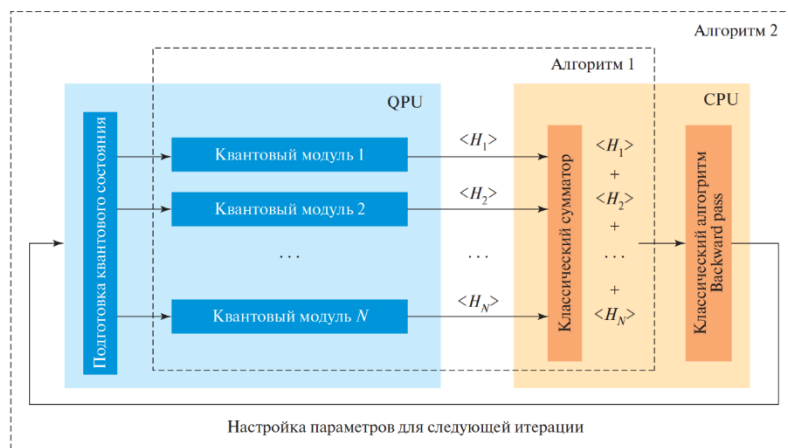


Рис. 14. Схема применения квантово-классических гибридных алгоритмов

Как показано на рис. 15 (а), обычно в *QNN* входные данные загружаются с использованием классических методов ввода данных.

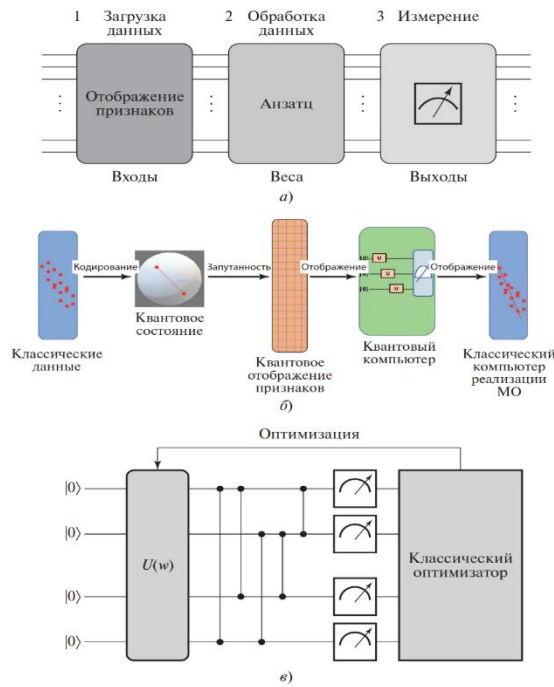


Рис. 15. (а) схема преобразования классических данных в квантовые данные; (б) отображение характеристик объектов в квантовой схеме; (в) пример квантовой нейронной сети

Квантовая схема содержит модуль отображения характеристик объектов с входными параметрами и модуль-анзац с настраиваемыми весами. Для получения выходных данных проводятся измерения.

Вариационные или параметризованные квантовые схемы (PQC) – особые типы квантовых алгоритмов, которые зависят от свободных параметров. PQC позволяют в полной мере использовать существующие квантовые компьютеры. В контексте QML PQC используются либо для кодирования данных, где параметры определяются кодируемыми данными, либо в качестве квантовой модели, где параметры определяются процессом оптимизации. PQC можно интерпретировать как модели ML, рассматривая вариационный квантовый классификатор, который использует две вариационные схемы (см. рис.15 (б)).

Схема состоит из трех основных компонентов, как показано на рис.16: операция инверсии матрицы (зеленый) предназначена для получения параметров гиперплоскости; оракул обучающих данных (синий) включен для подготовки состояния обучающих данных; и $U(x')$ предназначен для отображения тестового экземпляра x' в квантовые состояния.

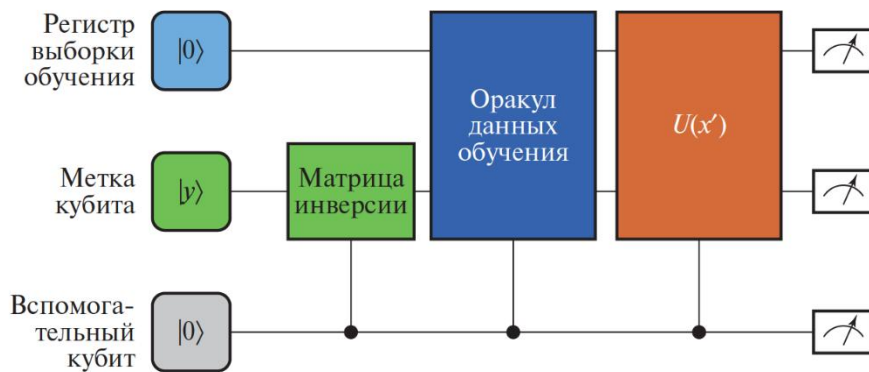


Рис. 16. Принципиальная схема квантовой SVM [11]

В классической *SVM* гиперплоскость получается путем минимизации функционала, в то время как в *qSVM* гиперплоскость получается путем решения линейных уравнений, что приводит к существенному ускорению.

Кубиты можно разделить на три группы: обучающие регистры (синие), которые представляют обучающие экземпляры, кубит меток (зеленый), который принимает метку, и вспомогательный кубит (серый). Для получения параметров гиперплоскости используется матричная инверсия. Затем применяется оракул обучающих данных для подготовки состояния обучающих данных. Классификация нового тестового экземпляра x' вводится с помощью операции $U(x')$.

Г. Квантовая кластеризация

Квантовая кластеризация – класс методов кластеризации данных, в которых используются математические и концептуальные инструменты квантовой механики. Он относится к семейству алгоритмов кластеризации на основе функции плотности, в которых кластеры определяются областями с более высокой плотностью точек данных. Его конструкция включает в себя масштабнo-пространственную вероятностную функцию, рассматриваемую в самом низком собственном состоянии уравнения Шредингера, за которой следуют простые аналитические операции для получения потенциальной функции, минимумы которой определяют центры кластеров (см. рис. 17).

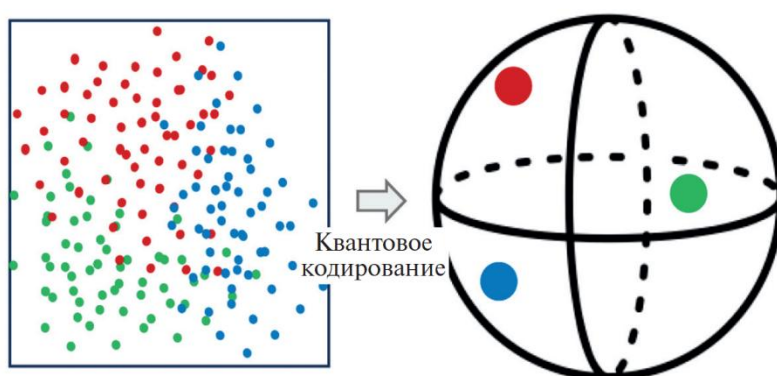


Рис. 17. Квантовая кластеризация (центры кластеров представлены в гильбертовом пространстве)

Примечание. Напомним, что в *ANN* (*artificial neural networks*) есть несколько слоев, представляющих нейроны. Более простые алгоритмы *ANN* не имеют обратной связи между слоями и называются нейронными сетями с прямой связью (*FFNN – feed-forward neural networks*). Однослойная нейронная сеть *FFNN* состоит из входного слоя нейронов и выходного слоя нейронов. В многоуровневой сети прямой связи первый уровень является входным уровнем, который принимает входной сигнал, а последний уровень является выходным уровнем. Между этими двумя слоями могут находиться несколько скрытых слоев. Сигнал от входного уровня проходит через эти скрытые слои к выходному слою. Связь между парой узлов (нейронами) в двух соседних слоях имеет соответствующий вес, который указывает на силу связи между ними. Входные данные для определенного слоя умножаются на вес, чтобы получить внутреннее значение, которое изменяется на пороговое значение перед передачей в функцию активации для получения выходных данных этого слоя. Эти выходные данные передаются следующему слою в качестве входных данных. Последний слой обеспечивает предоставление результатов работы сети. На каждой итерации веса и пороговые значения обновляются для получения более точного значения.

9. Квантовый перцептрон: качественные особенности

Мозг человека содержит примерно 200 миллиардов нейронов, и каждый нейрон состоит из четырех частей: дендритов, сомы, аксона и синапсов. Сигналы собираются нейронами через дендриты, а затем все собранные сигналы суммируются сомой. После достижения порогового значения сигнал передается другим нейронам через аксон. Мощность связи между нейронами определяется синапсами.

Аналогично, искусственная нейронная сеть (*ANN – artificial neural network* или иногда называемая просто *NN*) имитирует эту биологическую нейронную сеть. Таким образом, мозг можно рассматривать как нейронную сеть, организованную в чрезвычайно сложную структуру и состоящую из огромного количества ($\sim 10^{11}$) нейронов. На рис. 18 (а) показано существенно упрощенное изображение нейрона.

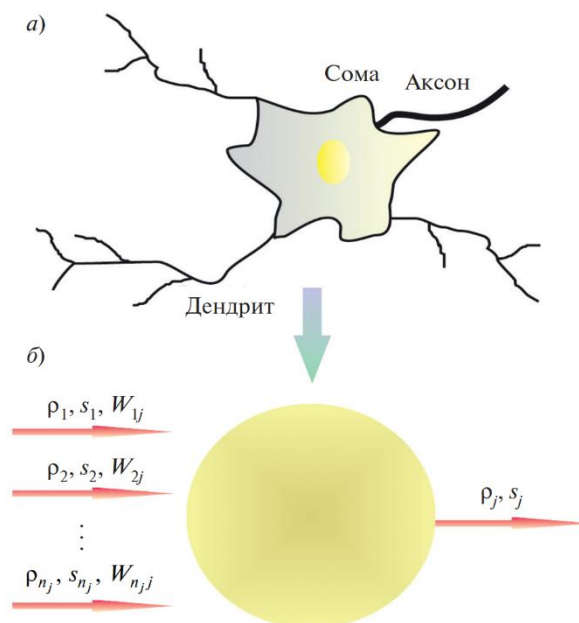


Рис. 18. Упрощенное изображение нейрона (а) и модель нейрона с «мягкой» квантовой структурой (б) [12]

[В (а) нейрон объединяет сотни или тысячи поступающих сигналов через свои дендриты. После обработки телом клетки нейрон передает сигнал через свой аксон другому нейрону для обработки в виде потенциала действия, когда его внутренний потенциал превышает определенный порог.]

Реальный нейрон объединяет сотни или тысячи поступающих сигналов через свои дендриты. Эти сигналы приводят к изменению внутреннего клеточного потенциала по сложной схеме, которая зависит от возбуждающей или тормозящей природы синапсов, через которые сигналы попадают в тело клетки. Нейрон передает сигнал через свой аксон другому нейрону для обработки в виде потенциала действия только тогда, когда его внутренний потенциал превышает определенный порог. У нейрона есть только один аксон, но существует набор дендритов, образующих древовидную структуру.

Вместо моделирования нейронов, как в обычных моделях нейронных сетей, или использования концепций, заимствованных из жестких квантовых вычислений, мы моделируем j -й нейрон внутри нейронной сети как базовую единицу «мягкого» квантового компьютера; сеть может иметь любую сетевую архитектуру. Как показано на рис.18 (б), в квантовой нейронной модели каждый нейрон имеет n_j входов s_i ($i = 1, 2, \dots, n_j$) от, возможно, n_j других нейронов, каждый из которых «взаимодействует» с j -м нейроном. Взаимодействие характеризуется супероператором $\mathcal{W}_{ij}(t)$ соединений вместо матрицы соединений (или взвешенных путей) w_{ij} в обычных нейронных сетях.

Примечание. Здесь уместны некоторые замечания.

1. Нейрон человеческого мозга обычно имеет в среднем около 10^4 синапсов. Именно такое большое количество входных нейронов активирует целевой нейрон. Согласно современной нейронауке, матрица соединений - математическая абстракция синаптической эффективности межнейронных синапсов, в то время как поступающие сигналы математически представляют собой потенциалы действия. В классической модели нейрона активация реального нейрона определяется внутренним интегрированным потенциалом клетки или математически сигнальной функцией и внутренним порогом возбуждения. Для реальных нейронов, поскольку потенциалы действия и потенциалы клеток изменяются намного быстрее, чем изменения синаптической эффективности, активация нейронов - быстрый процесс, в то время как изменения синаптической эффективности - медленный. В классических нейронных сетях изменение синаптической эффективности осуществляется с помощью «обучающего

алгоритма» в процессе обучения в ответ на стабильные паттерны активности. Однако в контексте текущей квантовой модели нейрона на приведенные выше замечания следует взглянуть под другим углом.

2. Здесь процесс обучения реализуется с помощью «квантового алгоритма обучения», то есть определенного временного паттерна квантовых операций $\mathcal{W}_{ij}(t)$, зависящего от состояний входных нейронов. Используя различные алгоритмы квантового обучения, квантовая нейронная модель открывает возможности для квантовых манипуляций с реальными нейронами, искусственным интеллектом и, в конечном счете, с человеческим мозгом. В отличие от этого, существующие квантовые нейронные сети моделируют нейроны как кубиты в чистом виде, а веса соединений - как обычные квантовые логические элементы. Однако человеческий мозг - в высшей степени открытая и декогерентная система. Очень маловероятно, что человеческий мозг можно смоделировать как хорошо изолированную квантовую систему. В модели человеческий мозг не является ни классическим компьютером (мягким или жестким), ни стандартным квантовым компьютером, а чем-то средним, а именно «мягким» квантовым компьютером. Моделируя нейронные узлы как кубиты в смешанных состояниях, а их соединения - как супероператоры соединений, нам остается увидеть, в какой степени наша модель может имитировать и понимать реальное функционирование человеческого мозга.

В математической формулировке квантовой модели нейрона n_j физических кубитов управляются n_j входами через супероператор связи $\mathcal{W}_{ij}(t)$. Как и в реальном нейроне, интегрирующем поступающие сигналы и выдающем сигнал, когда внутренний потенциал нейрона превышает определенный порог, здесь результирующее коллективное состояние n_j физических кубитов представляет собой интеграцию поступающих сигналов и определяет выходное состояние логического кубита. Таким образом, моделируем квантовый нейрон в каждом узле сети одновременно как n_j физических кубитов и как логический кубит, так что он может взаимодействовать с n_j поступающими сигналами и выдавать состояние одного кубита. Таким образом, предложенная выше нейронная модель максимально имитирует реалистичные нейроны и в то же время использует квантовые законы для обработки информации.

Математически мягкие квантовые вычисления начинаются с n двухуровневых квантовых систем (кубитов), связанных с окружающей их средой, и, таким образом, начальным состоянием n кубитов является смешанное состояние $\rho_{12\dots n}$ в вычислительном базисе $|0\rangle$ и $|1\rangle$. Эволюция такого мягкого квантового компьютера во время вычислений описывается полностью положительным отображением Крауса (обозначаемым супероператором \mathcal{O}): $\mathcal{O}(\rho_{12\dots n}) = \sum_k \hat{E}_k(t) \rho_{12\dots n} \hat{E}_k^\dagger(t)$. Здесь квантовые операции $\hat{E}_k(t)$, которые могут зависеть от времени, удовлетворяют равенству $\hat{E}_k(t) \hat{E}_k^\dagger(t) \leq I$, справедливому для отображения, сохраняющего след. Важными примерами операций, сохраняющих след, являются проективные измерения, унитарные эволюции и частичный след. После зашумленной эволюции квантовые измерения в вычислительном базисе завершают процесс мягких квантовых вычислений.

Мы можем моделировать нейроны как зашумленные кубиты; сеть таких зашумленных кубитов в условиях зашумленной эволюции и измерения - особая модель мягких квантовых вычислений.

Логический кубит изначально описывается матрицей плотности ρ_j^{in} . Моделируя каждый нейрон как логический кубит, мы рассматриваем только двоичные сигналы, а именно $s_i = 0$ или 1. Всякий раз, когда $s_i = 1$, что соответствует случаю ввода сигнала, с ρ_j^{in} будет выполняться квантовая операция (в данном случае супероператор) $\mathcal{W}_{ij}(t)$, в то время как при $s_i = 0$ (нулевом входном сигнале) с ρ_j^{in} ничего не произойдет. Очевидно, что такая условная операция, называемая далее «управляемой операцией» Крауса, является обобщением двухкубитных управляемых унитарных операций в жестких квантовых вычислениях. Между тем, каждый входной нейрон сам по себе с входным сигналом s_i также является логическим кубитом в смешанном состоянии ρ_i^{in} . В результате эволюция всей системы из исходного состояния $\otimes_{i=1}^{n_j} \rho_i^{in} \otimes \rho_j^{in}$ выглядит следующим образом:

$$\otimes_{\{i\}j}^{out} = \mathcal{T} \otimes_{i=1}^{n_j} \mathcal{O}_{ij} (\rho_i^{in} \otimes \rho_j^{in}) \equiv \mathcal{T} \otimes_{i=1}^{n_j} \left[\mathcal{P}_{|0\rangle_i} \otimes \hat{I}_j + \mathcal{P}_{|1\rangle_i} \otimes \mathcal{W}_{ij}(t) \right] (\rho_i^{in} \otimes \rho_j^{in}).$$

Здесь «суперпроекторы» $\mathcal{P}_{|s\rangle}$ определяются через $\mathcal{P}_{|s\rangle} \rho = |s\rangle \langle s| \rho |s\rangle \langle s|$, \hat{I} – оператор идентичности, а \mathcal{T} – упорядочение по времени. Вообще говоря, все $\mathcal{W}_{ij}(t)$ воздействуют на целевой нейрон с определенными временными паттернами, и поэтому важен временной порядок этих действий, поскольку разные $\mathcal{W}_{ij}(t)$ могут быть некоммутативными. В дополнение к условной динамической эволюции, каждый квантовый нейрон также может подвергаться локальным зашумленным операциям.

Суперпроекторы $\mathcal{P}_{|s\rangle}$ использовались для выбора вычислительного базиса в качестве входных данных для j -го нейрона. При этом фактически неявно предполагалось определенная декогеренция, вызванная окружающей средой, которая естественным образом возникает в организме клетки, чтобы выбрать предпочтительную основу, а именно вычислительную основу. Чтобы быть последовательными для квантовой модели нейрона, необходимо определить выходной сигнал s_j j -го нейрона таким же образом. Конечное состояние j -го нейрона может быть получено путем вычисления следа от всех входных состояний, а именно, $\rho_j^{out} = \text{Tr}_{\{i\}} \rho_{\{i\}j}^{out} = \mathcal{T} \prod_{i=1}^{n_j} \left[p_i \hat{I}_j + (1 - p_i) \mathcal{W}_{ij} \right] \rho_j^{in}$, где $p_i \equiv p_i(0) = \text{Tr}(|0\rangle_i \langle 0| \rho_i^{in})$ – вероятность $|0\rangle_i$ в пределах действия ρ_i^{in} . Тогда выходное состояние равно $\rho_j = \left(\mathcal{P}_{|0\rangle_j} + \mathcal{P}_{|1\rangle_j} \right) \rho_j^{out}$. Другими словами, выходной сигнал s_j j -го нейрона равен

$$s_j = \begin{cases} 0 & \text{с вероятностью } p_j(0) \\ 1 & \text{с вероятностью } 1 - p_j(0) \end{cases}$$

Это завершает описание модели квантового нейрона для реализации на «мягком» квантовом компьютере. Приведенные выше требования к входному и выходному состояниям в предпочтительной вычислительной базе предполагают, что каждый нейрон в модели является квантовым само-измеряющимся. Это устраняет неоднозначность обозначения сигнальных функций различной формы в классических нейронных сетях.

Пример: простейший случай с двумя нейронами. Для двух нейронов в начальных состояниях $\rho_1^{in} = p_1 |0\rangle_1 \langle 0| + (1 - p_1) |1\rangle_1 \langle 1|$ ($p_1 \neq 0, 1$) и ρ_2^{in} действие контролируемой операции Крауса \mathcal{O}_{12} приводит к конечному состоянию

$$\rho_{12}^{out} = \left(\mathcal{P}_{|0\rangle_1} \otimes \hat{I}_2 + \mathcal{P}_{|1\rangle_1} \otimes \mathcal{W}_{12} \right) (\rho_1^{in} \otimes \rho_2^{in}) = p_1 |0\rangle_1 \langle 0| \otimes \rho_2^{in} + (1 - p_1) |1\rangle_1 \langle 1| \otimes \mathcal{W}_{12} (\rho_2^{in}),$$

где \mathcal{W}_{12} представляет собой определенный квантовый канал. Имеет ли ρ_{12}^{out} неклассическую (или квантовую) корреляцию? Квантовые корреляции ρ_{12}^{out} , если таковые имеются, могут быть количественно определены с помощью квантового беспорядка. Любое двух-частичное состояние называется полностью классически коррелированным, если оно имеет вид

$$\rho_{12} = \sum_{i,j} p_{ij} |i\rangle_1 \langle i| \otimes |j\rangle_2 \langle j|;$$

в противном случае имеем квантовую корреляцию. Здесь $|i\rangle_1$ и $|j\rangle_2$ – ортонормированные базисы двух частиц с неотрицательными вероятностями p_{ij} .

Очевидно, что в приведенном уравнении для двухнейронного состояния ρ_{12}^{out} первый нейрон становится коррелированным с неортогональными состояниями второго нейрона, поскольку $\mathcal{W}_{12}(\rho_2^{in})$ и ρ_2^{in} неортогональны, а именно, ρ_{12}^{out} имеет квантовые корреляции.

На самом деле, ρ_{12}^{out} — это классическое квантовое состояние. В то время как для измерений на первом нейроне беспорядок равен нулю, измерения на втором нейроне в целом приводят к ненулевому беспорядку.

Таким образом, нейронная сеть может создавать квантовые корреляции, хотя доступны только смешанные начальные состояния и осуществляются операции с очень большим количеством помех. Следовательно, если мягкие квантовые вычисления действительно имитируют рабочий механизм человеческого мозга, то мозг, безусловно, является квантовым инструментарием, поскольку между нейронами в нем существуют квантовые корреляции.

Несмотря на то, что в настоящее время доступны прототипы реальных квантовых компьютеров, они по-прежнему имеют ограниченный размер и сильно подвержены воздействию шума, что создает дополнительную проблему для исследователей в этой области и приводит к возникновению так называемой квантовой эры промежуточного масштаба с шумами (*NISQ*). Вариационные квантовые алгоритмы (*VQA*) – особый подход *QML*, который продемонстрировал свою высокую эффективность в работе с таким зашумленным оборудованием, используя комбинацию параметризованных квантовых схем (*PQC*), часто называемых также квантовыми нейронными сетями (*QNN*), и процесс оптимизации выполняется на классическом компьютере.

Из-за их сходства с классическими *NN*, *PQC*, используемые в *VQA*, часто называются *QNN*. Они состоят из этапа кодирования данных, параметризованного анализа и операции измерения в конце для получения результата, за которым следует классическое обновление параметров. Следуя этой схеме, *QNN* могут научиться выполнять различные задачи, такие как регрессия и классификация.

Классические паттерны $x \in \mathbb{R}^n$ кодируются в квантовой схеме из N кубитов с помощью квантовой схемы, являющейся квантовым отображением признаков, и соответствует унитарной матрице $S(x)$, отображающей x в 2^N -мерное гильбертово пространство. Затем за вложением следует вариационный анзац $W(\theta)$. Последний состоит из слоев параметризованных однокубитных гейтов вращения и двухкубитных гейтов формирования запутанных состояний: гейты вращения используются для манипулирования квантовым состоянием путем независимого обращения к отдельным кубитам, в то время как слои из гейтов запутанных состояний позволяют создавать многостороннюю запутанность по всему вектору состояний. Это можно резюмировать следующим образом:

$$W(\theta)S(x)|0\rangle = W(\theta)|\phi(x)\rangle, \text{ где } |0\rangle = |0\rangle^{\otimes N}.$$

Для реализации более содержательных моделей при повторной загрузке данных могут быть повторно применены кодирование данных и унитарный анзац. В связи с этим, учитывая L уровней повторной загрузки данных, итоговый унитарный оператор, описывающий эволюцию квантового состояния, формально выражается как

$$U_L(x; \theta) = \prod_{l=L}^1 U_l(x; \theta) = \prod_{l=L}^1 W(\theta^{(l)})S(x),$$

где $\theta^{(l)}$ относится к параметрам в l -м слое квантовой схемы.

Квантовые гейты вращения играют роль искусственных нейронов, а гейты запутанных состояний работают как соединения между ними. Нейроны обрабатывают поступающую информацию (от предыдущего слоя нейронов/квантовой схемы) и затем отправляют ее всем нейронам следующих уровней по всем их соединениям; для *QNN* это всего лишь схематическое представление; на самом деле, если состояние запутано, вращение на одном кубите также повлияет на другие состояния. В классической *NN*, когда один нейрон переобучен, он удаляется вместе со всеми своими связями, тогда как при каноническом квантовом переобучении некоторые гейты запутанных состояний элементы остаются неизменными, чтобы избежать сильного влияния на процесс квантового обучения.

В конце квантовой схемы выполняется измерение относительно наблюдаемого объекта для получения результатов работы алгоритма: $f_L(x; \theta) = \langle 0|U_L^\dagger(x; \theta)\hat{O}U_L(x; \theta)|0\rangle$, где \hat{O} – оператор измерения. Обычно измеряем в базисе Паули-*Z* один или несколько кубитов в зависимости от задачи. При

использовании в VQA классический оптимизатор из $f_L(x; \theta)$ вычисляет функцию затрат для обновления параметров.

Таким образом, квантовые нейронные сети - вариационные модели, которые вычисляют математическое ожидание для оптимизации функции затрат.

Пример QNN представлен на рис. 20.

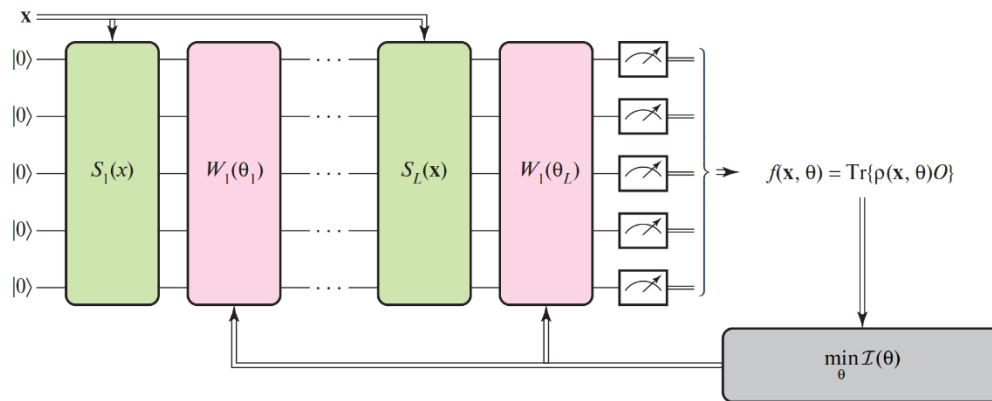


Рис. 20. Прототип архитектуры QNN [13]

Классические данные преобразуются в квантовые состояния с помощью операций кодирования $S_i(x)$. Затем для обработки данных применяются параметризованные квантовые операции $W_i(\theta)$. Часто эта структура повторяется для нескольких уровней. После последнего слоя выполняется измерение и вычисляется ожидаемое значение f , затем параметры θ оптимизируются с помощью классического оптимизатора для минимизации подходящей эмпирической функции потерь \mathcal{L} .

Чтобы обеспечить квантовую обработку, данные кодируются в квантовом состоянии

$$|\psi(x, \theta)\rangle = U(x, \theta)|0\rangle.$$

На практике оператор кодирования $U(x, \theta)$ реализуется квантовой схемой, которая манипулирует начальным квантовым состоянием n -кубита $|0\rangle$ на основе данных $x \in \mathbf{X}$. Хотя были предложены различные варианты кодирования, такие как кодирование данных непосредственно в амплитуды волновой функции или побитно в базовые состояния, преобладающий подход для алгоритмов, совместимых с NISQ, основан на кодировании компонентов x_i векторов данных x в виде углов в одном или двух направлениях - вращения кубитов. Как правило, эти операции дополняются логическими элементами, независимыми от данных, параметризованными некоторыми параметрами $\theta = (\theta_1, \dots, \theta_k)$, где $\theta_i \in \mathbb{R}$. Полученную схему часто называют схемой кодирования.

Схема кодирования $U(x, \theta)$ часто представляют многоуровневой структурой $U(x, \theta) = \prod_{i=1}^L S_i(x)W_i(\theta_i)$, где $S_i(x)$ обозначает операцию, зависящую от данных, $W_i(\theta_i)$ - параметризованные операторы, а L - количество слоев. Результирующее состояние $\rho(x, \theta)$ многократно измеряется для вычисления математического ожидания некоторого наблюдаемого значения O в виде $f(x, \theta) = \text{Tr}(\rho(x, \theta)O)$. Параметры θ оптимизируются за счет минимизации эмпирических потерь

$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(x_i, y_i, \theta)$, где θ - подходящая функция потерь, такая как евклидово расстояние между $f(x_i)$ и y_i .

Пример: модель квантового нейрона. В ряде работ описывается совместное моделирование искусственных нейронов с использованием как классических, так и квантовых алгоритмов. Заметный недавний вклад в эту область внесли Таккино и соавторы, которые продемонстрировали, что искусственный

нейрон, способный анализировать паттерны в двоичных изображениях, может быть реализован на реальном квантовом компьютере: одном из квантовых процессоров IBM. Входными данными для алгоритма является вектор $\vec{x} \in \{-1, 1\}^N$, $N = 2^q$, который преобразуется в следующую волновую функцию:

$$|\psi_{\vec{x}}\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N-1} x_j |j\rangle, \text{ где } N - \text{размерность входной строки, } x_j - \text{значение } j\text{-го входного сигнала и } |j\rangle$$

представляет состояния $|j\rangle \in \{|00\dots 00\rangle, |00\dots 01\rangle, \dots, |11\dots 11\rangle\}$, которые формируют базисные векторы для q кубитов. Из-за трудностей с реализацией они смогли смоделировать только $q = 2$ кубита на квантовом компьютере.

Преимущество подхода заключается в том, что это позволяет более точно контролировать динамику системы за счет снижения частоты ошибок. Аналогичным образом, вектор веса $\vec{w} \in \{-1, 1\}^N$ сопоставляется с состоянием $|\psi_{\vec{w}}\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N-1} w_j |j\rangle$. Следующий шаг включает в себя вычисление внутреннего произведения между \vec{w} и \vec{x} . Сначала унитарная матрица U_w поворачивает $|\psi_{\vec{w}}\rangle$ к оси $|11\dots 11\rangle$, т.е. $U_w |\psi_{\vec{w}}\rangle = |1\rangle^{\otimes q} = |N-1\rangle$. Последующее применение U_w к $|\psi_{\vec{x}}\rangle$ соответствует проекции вектора $U_w |\psi_{\vec{x}}\rangle$ на ось $|11\dots 11\rangle$. Отсюда следует, что скалярное произведение между двумя квантовыми состояниями равно $\langle \psi_{\vec{w}} | \psi_{\vec{x}} \rangle = \langle \psi_{\vec{w}} | U_w U_w^\dagger | \psi_{\vec{x}} \rangle = \langle N-1 | \phi_{x,w} \rangle = c_{N-1}$, где c_{N-1} - коэффициент последнего состояния $|\phi_{x,w}\rangle$ и $|\phi_{x,w}\rangle = U_w^\dagger |\psi_{\vec{x}}\rangle$. Чтобы вычислить скалярное произведение между \vec{w} и \vec{x} , уравнения объединяются таким образом, что:

$$\langle \psi_{\vec{w}} | \psi_{\vec{x}} \rangle = \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \langle j | \right) \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} w_j | j \rangle \right) = \frac{1}{N} \sum_{j=0}^{N-1} x_j w_j \langle j | j \rangle = \frac{1}{N} \vec{w} \cdot \vec{x}.$$

Таким образом, получаем скалярное произведение $\vec{w} \cdot \vec{x} = N \langle \psi_{\vec{w}} | \psi_{\vec{x}} \rangle$. Поскольку $\langle \psi_{\vec{w}} | \psi_{\vec{x}} \rangle = c_{N-1}$ нормализованное скалярное произведение может быть извлечено из последнего состояния $|\phi_{x,w}\rangle$. Это может быть реализовано в схеме, использующей вспомогательный кубит и мультиуправляемый элемент NOT, который изменяет состояние вспомогательного кубита в зависимости от состояния самого кубита (рис.21).

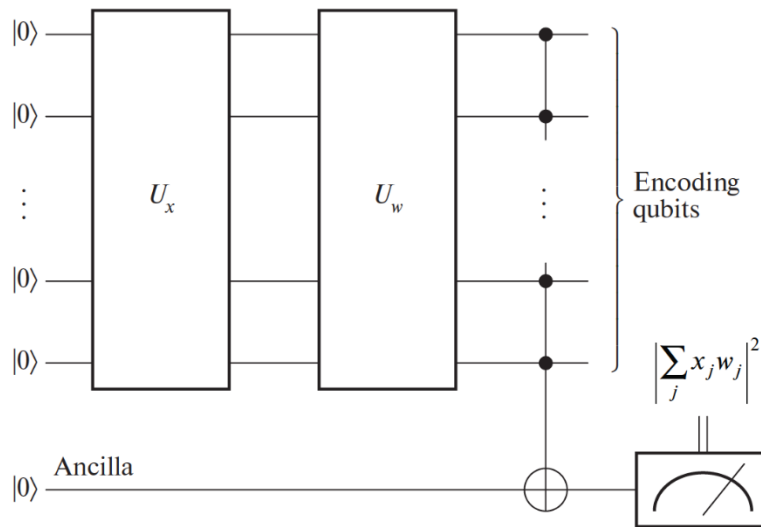


Рис.21. Схема реализации квантового нейрона [14]

[Скалярное произведение между \vec{w} и \vec{x} вычисляется и сохраняется как коэффициент c_{N-1} последнего кубита входных данных. Эта информация передается во вспомогательный кубит с помощью мультиуправляемого вентиля CNOT. Затем измеряется вспомогательный кубит; он коллапсирует в $|1\rangle$ с вероятностью, пропорциональной c_{N-1} или $|0\rangle$ в противном случае. Процесс измерения имитирует нелинейность.]

В частности, это приведет только к переводу вспомогательного кубита $|1\rangle$ в конечное состояние $|\phi_{x,w}\rangle$, в котором c_{N-1} хранится. После применения многоуправляемого элемента NOT состояние системы будет следующим: $|\phi_{x,w}\rangle|0\rangle \rightarrow \sum_{j=0}^{N-2} c_j |j\rangle|0\rangle + c_{N-1} |N-1\rangle|1\rangle$. Нелинейность определяется путем измерения вспомогательного кубита. Это приведет к коллапсу кубита в $|1\rangle$ с вероятностью $|c_{N-1}|^2$.

Следовательно, выходные данные системы являются вероятностными $P(y=1) = \left(\frac{\vec{w} \cdot \vec{x}}{N}\right)^2$. Значение этой вероятности также может быть оценено путем многократной выборки выходных данных и вычисления частоты $y=1$.

Примечание. В процессе обучения у этого метода есть потенциальный недостаток. Предположим, что базовая метка истинности равна 0, но алгоритм предсказывает, что коэффициент больше 0,5, и ошибочно классифицирует его как 1. Чтобы обновить параметры веса для следующей итерации, алгоритм выполняет поиск случаев, когда \vec{w} и \vec{x} имеют соотношение $\vec{w} \cdot \vec{x}$, что привело к увеличению внутреннего произведения. Однако, чтобы уменьшить вероятность того, что алгоритм выдаст значение 1, требуется, чтобы внутреннее произведение уменьшалось.

Таким образом, определяется подмножество, в котором они совпадают, и случайным образом меняют местами некоторую долю битов в \vec{w} . Проблема в этой задаче возникает, когда очень большое подмножество \vec{w} и \vec{x} не совпадают, поскольку внутреннее произведение является большим отрицательным значением, и, следовательно, квадрат внутреннего произведения является большим положительным значением. Теперь, когда алгоритм выполняет поиск мест, где они совпадают, очень немногие значения совпадают. Таким образом, при выборе любого из них внутреннее произведение становится более отрицательным, что делает квадрат более положительным, и поэтому вероятность того, что результат будет равен 1, выше, что противоположно тому, что должно произойти.

Общее представление о схеме можно увидеть на рис.21. Несмотря на то, что модель успешно реализована в процессоре IBM для классификации шаблонов, время, необходимое для запуска алгоритма, было линейным по размерности входного вектора, что соответствует классическим моделям. Таким образом, у модели нет преимущества во времени по сравнению с тем, что существует в настоящее время. Однако преимущество перед классическими методами получают в плане сохранения параметров во время обучения, кодируя свои данные в экспоненциально меньшем количестве кубитов.

При сохранении модели в памяти это преимущество исчезло бы, поскольку пространство, необходимое для хранения весов, является линейным; однако, если бы память была узким местом во время обучения (например, все параметры не могли поместиться в *qRAM*), этот метод обеспечил бы преимущество.

10. Квантовый вариационный перцептрон

Квантовый вариационный перцептрон (*QVP – Quantum Variational Perceptron*) - модель машинного обучения, которая сочетает в себе квантовые вычисления и искусственные нейронные сети. Реализация метода QVP основана на разработке модели вариационной квантовой схемы (*VQC – variational quantum circuit*), которая показывает, как квантовые схемы могут представлять параметры нейронной сети. QVP использует квантовые схемы для выполнения сложных вычислений, которые трудно выполнить классическими методами, а классический оптимизатор настраивает параметры этих квантовых схем, чтобы сделать их оптимальными.

Одним из главных преимуществ QVP является то, что он может обрабатывать многомерные и нелинейные данные, которые трудно обрабатывать с помощью классических нейронных сетей. QVP также может выполнять квантово-расширенное извлечение признаков для повышения точности модели, что важно для обработки естественного языка, поиска лекарств и классификации. QVP также обладает потенциальными преимуществами перед классическими нейронными сетями с точки зрения устойчивости к шуму и обобщения, что покажем ниже. Устройства квантовых вычислений чувствительны к шуму, но QVP может использовать методы квантовой коррекции ошибок для устранения этой проблемы. Такая устойчивость к помехам делает QVP подходящим для приложений, где данные являются неопределенными или зашумленными.

Цель состоит в создании возможности использовать квантовые вычисления для определения функции $f(x; \theta)$, а затем использовать классический оптимизатор для оптимизации функции потерь, поэтому представим задачу с помощью квантовой схем. Схема разрабатывается в три этапа (см., рис. 15 (а,б)): подготовка состояния, квантовая схема модели и измерение.

А. Подготовка квантового состояния

Подготовка состояния является важным этапом в квантовой вариационной схеме. Она включает в себя подготовку исходного квантового состояния в качестве входных данных для остальной части разрабатываемой схемы, и можно определить это состояние с помощью набора параметров, которые изменяем для оптимизации выходных данных схемы. Существует множество различных подходов к подготовке состояния, но один из популярных методов заключается в использовании последовательности унитарных преобразований для преобразования известного начального состояния в требуемое вариационное состояние. Эти унитарные преобразования могут быть реализованы с использованием различных квантовых элементов, таких как элементы вращения, элементы управляемой НЕТ – последовательности схем и элементы преобразования фазы. Важно тщательно выбирать начальное состояние и унитарные преобразования, чтобы максимизировать вероятность получения желаемого результата от квантовой схемы.

Итак, первый шаг - закодировать классические данные x в квантовые данные; как уже упоминали, для этого существуют различные подходы, такие как базовое кодирование, угловое кодирование, кодирование высшего порядка и амплитудное кодирование.

Каждая методология имеет индивидуальное применение. Например, в методе базисного кодирования квантовое состояние представляется на определенной основе, включая вычислительный базис или базис Фурье. И наоборот, в методе углового кодирования квантовое состояние выражается в терминах углов поворота, примененных к состоянию. Кроме того, при кодировании более высокого порядка квантовое состояние представляется с использованием свойств более высокого порядка, таких как запутанность или квантовая когерентность.

В нашем случае выбираем метод амплитудного кодирования, который означает, что данные напрямую связаны с амплитудой квантового состояния; идея состоит в том, чтобы представить эти квантовые состояния в виде комплексно-значных амплитуд, чтобы описать вероятность измерения конкретного результата при измерении квантового состояния. В этом методе квантовые состояния представляем в виде векторов в сложном векторном пространстве, а квантовые операции - в виде матриц, действующих на эти векторы. Это представление предлагает компактную и интуитивно понятную структуру для описания квантовых состояний и операций, что делает его особенно полезным в контексте квантовых вариационных схем. Преимуществом этого метода является возможность кодирования набора данных из M входных данных с помощью N функций, для которых требуется только $n = \log(N \cdot M)$ кубитов; он основан на создании оператора $\phi(x)$, который приведет к состоянию $\phi(x)|0_p\rangle$ с количеством p - кубитов.

Б. Модель квантовой схемы

Этап моделирования квантовой схемы является важнейшим компонентом процесса оптимизации. На этом этапе для моделирования квантового состояния создается параметризованная квантовая схема, называемая «анзатцем», в форме унитарного оператора. Параметры схемы настраиваются итеративно, чтобы свести к минимуму разницу между состоянием модели и целевым состоянием. Этап моделиро-

вания схемы важен для достижения точной и эффективной подготовки квантового состояния и реализации алгоритма. Кроме того, производительность и успех квантовой схемы зависят от того, насколько правильно выбрана предварительно структура «анзатца» и алгоритм оптимизации.

Итак, построим $\mathcal{U}(\theta)$, унитарный оператор из квантового состояния $|\phi(x)\rangle$, который представляет вектор x в квантовой схеме, так что $\mathcal{U}(\theta)|\phi(x)\rangle$ – это квантовое состояние, которое можно измерить, а θ – обучаемый параметр. После этого разложили $\mathcal{U}(\theta) = \mathcal{U}_1 \dots \mathcal{U}_L$ как произведение 2-кубитных параметризованных элементов управления кубитами, затем используем комбинацию общих унитарных элементов управления и элементов *CNOT*; в результате получили выражение \mathcal{U} следующим образом:

$$\mathcal{U}(\theta, \phi, \lambda) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} \sin\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i\lambda+i\phi} \cos\left(\frac{\theta}{2}\right) \end{bmatrix}.$$

В. Измерение

Этап измерения позволяет извлекать информацию из квантового состояния, он включает в себя проекцию квантового состояния на измерительную основу, а сбор результирующих результатов измерений используется для вычисления функции потерь; используем эту функцию для настройки параметров квантовой схемы на этом текущем этапе. Итак, начинаем с генерации первого кубита квантового состояния, используя схему, показанную на рис. 22.

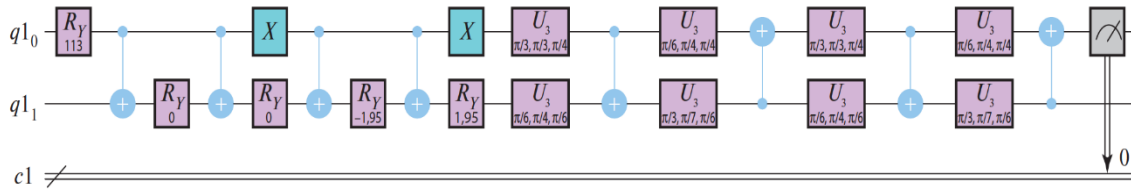


Рис. 22. Схема квантового вариационного перцептрона (Quantum Variational Perceptron - QVP)

Затем измеряем его, и это дает следующее выражение:

$$f(x; \theta) = P(y|x=1; \theta) = P(q_0 = 1|x; \theta) = \sum_{k=1}^n \left| (\mathcal{U}(\theta)\phi(x))_k \right|^2.$$

Модель обучена минимизировать функцию потерь, которая может быть определена следующим образом: $\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N l(\pi(x_i; \theta), y_i) \pi(x; \theta)$. Упростим приведенное выражение, используя $\pi(x; \theta)$ как $P(q_0 = 1|x; \theta)$. Параметры обновляются с помощью пакетного стохастического градиентного спуска; сложность заключалась в точном вычислении градиента. Однако, можно оценить градиент с помощью квантовых схем, и в результате получаем уравнение

$$\nabla_{\theta} \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} (l(\pi(x_i; \theta), y_i)) = \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} \pi(x_i; \theta) \partial_l l(\pi(x_i; \theta), y_i),$$

где $\partial_l l$ - частная производная от l по первой переменной, а выражение потери градиента имеет вид: $\nabla_{\theta} \pi(x_i; \theta) = -\frac{1}{2} \nabla_{\theta} \mathbb{E}(\sigma_z) = -\frac{1}{2} \nabla_{\theta} \langle \phi(x) \mathcal{U} | \sigma_z | \mathcal{U} \phi(x) \rangle$ с помощью $\sigma_z \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I}$ сокращенно обозначается как σ_z . Рассмотрев u как элемент вектора θ , можем представить $\partial_u \pi(x; \theta)$ в

виде: $\partial_\nu \pi(x; \theta) = -\frac{1}{2} \nabla_\theta \langle \phi(x) \partial_\nu \mathcal{U} | \sigma_z | \mathcal{U} \phi(x) \rangle - \frac{1}{2} \nabla_\theta \langle \phi(x) \mathcal{U} | \sigma_z | \partial_\nu \mathcal{U} \phi(x) \rangle$ и, используя правила вывода взятия производных, приведенное уравнение стало равным приведенному ниже выражению

$$\partial_\nu \pi(x; \theta) = -\frac{1}{2} \nabla_\theta \left(\langle \phi(x) \partial_\nu \mathcal{U} | \sigma_z | \mathcal{U} \phi(x) \rangle + \langle \phi(x) \partial_\nu \mathcal{U} | \sigma_z | \mathcal{U} \phi(x) \rangle^* \right) = -\operatorname{Re} \left\{ \langle \phi(x) \partial_\nu \mathcal{U} | \sigma_z | \mathcal{U} \phi(x) \rangle \right\}$$

Поскольку $\partial_\nu \mathcal{U}$ не является унитарным оператором, мы не можем использовать его в квантовой схеме; однако достигаем этого благодаря разработанной схеме, потому что производная от \mathcal{U} относится только к производной от элементарного элемента, что означает, что $\partial_\nu \mathcal{U} = \mathcal{U}_1 \dots \partial_\nu (\mathcal{U}_i) \dots \mathcal{U}_L$, где \mathcal{U}_i - это элемент, которому принадлежит ν .

Для элементарного унитарного элемента, как описано выше, имеем следующие тождества:

$$\partial_\theta \mathcal{U} = \frac{1}{2} \mathcal{U}(\theta + \pi, \phi, \lambda), \quad \partial_\phi \mathcal{U} = \frac{i}{2} (\mathcal{U}(\theta, \phi, \lambda) - \mathcal{U}(\theta, \phi + \pi, \lambda)), \quad \partial_\lambda \mathcal{U} = \frac{i}{2} \mathcal{U}(\theta, \phi, \lambda) - \mathcal{U}(\theta, \phi, \lambda + \pi)$$

Следовательно, $\partial_\nu \mathcal{U}$ может быть вычислен в виде:

$$\sum_{k=1}^K a_k \operatorname{Re} \left\{ \langle \phi(x) \mathcal{U}(\theta^{[k]}) | \sigma_z | \mathcal{U} \phi(x) \rangle \right\} + \sum_{l=1}^L b_l \operatorname{Im} \left\{ \langle \phi(x) \mathcal{U}(\theta^{[l]}) | \sigma_z | \mathcal{U} \phi(x) \rangle \right\},$$

где $\theta^{[k]}$ и $\theta^{[l]}$ - модифицированный вектор параметров из приведенных выше тождеств, мнимая часть берется из i в тех же тождествах для ϕ и λ , в результате чего можем вычислить функцию потерь.

Г. Квантовый вариационный перцептрон с алгоритмом Гровера

Познакомившись с основными свойствами кубитов и тем, как применять к ним операции, можно интуитивно понять, как квантовые вычисления могут привести к ускорению вычислений при правильном использовании. Предположим, у нас есть n кубитов. Квантовое состояние $|\phi\rangle$ этих кубитов может быть записано в виде линейной комбинации $r \leq 2^n$ линейно независимых векторов $|\phi_i\rangle \in \mathbb{C}^{2^n}$, таких, что $|\phi\rangle = \sum_{i=1}^r \alpha_i |\phi_i\rangle$. Теперь, применяя унитарную операцию U к квантовому состоянию $|\phi\rangle$, получаем $U|\phi\rangle = \sum_{i=1}^r \alpha_i U|\phi_i\rangle$, используя базовую линейную алгебру. Это означает, что квантовые вычисления позволяют параллельно выполнять r различных операций над r различными состояниями $|\phi_i\rangle$, причем число r потенциально растет экспоненциально с увеличением числа n кубитов. Это часто называют «квантовым параллелизмом». Поскольку при измерении найдем только одно базовое вычислительное состояние, этот параллелизм не сразу окажется полезным. Более точно, квантовый алгоритм должен состоять из квантовых операций, которые с очень высокой вероятностью приводят к конечному состоянию желаемого решения, такому, которое может быть найдено при измерении. Используя эти свойства квантовых компьютеров, можно разработать квантовые алгоритмы, которые будут решать задачи потенциально более эффективно, чем алгоритмы для классических компьютеров.

Пример: Алгоритм поиска Гровера - особенности вычислительного процесса. Одним из квантовых алгоритмов, который обеспечивает ускорение по сравнению с его классическими аналогами, является алгоритм поиска Гровера. Алгоритм Гровера был впервые описан в 1996 году и по сей день остается важной частью образования в области квантовых технологий. Это связано с тем, что это относительно простой алгоритм, который элегантно использует потенциал квантовых вычислений. Несмотря на то, что алгоритм Гровера может быть относительно простым, начинающие могут сосредоточиться на понимании математики и упустить из виду этапы алгоритма, которые реализуются с помощью квантовой механики. В данном разделе приводится «педагогическая» модель для интуитивного объяснения этих квантовых шагов в алгоритме Гровера, которая не требует глубоких знаний лежащей в его основе математики. Таких шагов три, и мы описываем их так, что каждый из них использует определенный ключевой принцип квантовой механики. Позже используем это, чтобы показать, как описанная «педагогическая» модель лучше справляется с внедрением этих принципов, чем существующие модели.

Проблема поиска. Задача поиска, решаемая алгоритмом Гровера, в терминах информатики заключается в поиске в неструктурированной базе данных: задача состоит в том, чтобы найти элементы из области поиска, которые обладают определенным свойством. Например, это могут быть элементы, удовлетворяющие набору математических уравнений. Эта задача поиска может быть переведена на квантовый язык как задача поиска в большом пространстве \mathbb{C}^{2^n} , охватываемом (натянутыми) 2^n базисными состояниями, тех базисных состояний $|i\rangle$, которые представляют элементы с желаемым свойством.

Задача поиска в алгоритме Гровера определяется в терминах функции оракула. Такая функция квантового оракула может распознавать правильные состояния $|i\rangle$ в большом наборе всех квантовых базовых состояний: при вводе общего квантового состояния $|\phi\rangle$ в качестве входных данных функция оракула распознает базовые состояния $|i\rangle$ с желаемым свойством и помечает их знаком минус. Здесь важно отметить, что оракул на самом деле не «знает», какие базовые состояния $|i\rangle$ представляют элементы с желаемым свойством; он может только распознать, является ли базовое состояние «хорошим», а какое базовое состояние «плохим», при вводе входных данных.

Как правило, распознать правильное решение проще, чем знать его заранее – например, легче проверить, удовлетворяет ли решение уравнению, чем вывести решение для этого уравнения. Мы обозначим этот квантовый оракул через O_X , где X - набор индексов i базовых состояний $|i\rangle$, которые представляют элементы с желаемым свойством. Следовательно, этот квантовый оракул зависит от проблемы. Таким образом, функция оракула является единственной частью квантовой схемы, реализация которой будет отличаться в зависимости от задачи, остальная часть схемы алгоритма Гровера универсальна.

Примечание. Задачу поиска в неструктурированной базе данных можно сравнить с поиском в лабиринте пути, ведущего к выходу из лабиринта, и мы используем это сравнение в нашей «педагогической» модели для объяснения алгоритма Гровера. Большинство путей в лабиринте не ведут к выходу, но некоторые пути обладают этим особым свойством, и нам нужен один из них. Из этого примера можно лучше понять разницу между знанием и распознаванием правильного решения: узнать, какие пути в лабиринте ведут к выходу, гораздо сложнее, чем просто определить, ведет ли данный путь к выходу.

Описание алгоритма Гровера. Алгоритм Гровера можно разделить на три этапа. Первый можно интерпретировать как создание состояния, охватывающего все элементы в базе данных. Второй этап - поиск «хороших» элементов с помощью функции оракул (oracle). Затем на третьем этапе выявляется один из этих «хороших» элементов.

Шаг 1: распространение по всему пространству \mathbb{C}^{2^n} Мы начинаем с кубитов в состоянии $|0\rangle$, и первое, что мы хотим сделать, - это распространить наше квантовое состояние на все пространство \mathbb{C}^{2^n} . Мы можем сделать это, применив операцию, которая является стандартной в квантовых вычислениях: применив матрицу Адамара H ко всем n кубитам. Эта операция дает нам новое состояние, называемое $|\mathcal{U}\rangle$ (суперпозиция) вида

$$|\mathcal{U}\rangle = H^{\otimes n} |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

Это означает, что кубиты приводятся в состояние, представляющее собой равную суперпозицию всех базовых состояний, и, таким образом, распределяются по всему пространству \mathbb{C}^{2^n} . Находясь в этом состоянии $|\mathcal{U}\rangle$, мы с уверенностью знаем, что все «хорошие» базовые состояния, соответствующие решению нашей поисковой задачи, также присутствуют в квантовом состоянии n кубитов.

Таким образом, можем записать это состояние $|\mathcal{U}\rangle$ как линейную комбинацию «хороших» и «плохих» состояний. Суперпозиция «хороших» состояний равна $|\mathcal{G}\rangle = \frac{1}{\sqrt{\dim(X)}} \sum_{i \in X} |i\rangle$, а суперпозиция «плохих» состояний равна $|\mathcal{B}\rangle = \frac{1}{\sqrt{2^n - \dim(X)}} \sum_{i \notin X} |i\rangle$.

Тогда получим: $|\mathcal{U}\rangle = \sin(\theta)|\mathcal{G}\rangle + \cos(\theta)|\mathcal{B}\rangle$, где $\theta = \arcsin(\sqrt{\dim(X)/2^n})$.

На этом первом шаге алгоритм Гровера создает состояние, представляющее собой линейную комбинацию всех базовых состояний. В терминах метафоры лабиринта это означает, что после шага 1 Гровер выдает состояние, описывающее все возможные пути в лабиринте: «хорошие», которые ведут к выходу, и «плохие», которые этого не делают.

Шаг 2: повторение итераций алгоритма Гровера и визуализация решений. Распределив квантовое состояние так, чтобы оно находилось в равной суперпозиции всех возможных базовых состояний, мы используем функцию «oracle», чтобы «отметить» «хорошие» базовые состояния, которые представляют элементы с искомым желаемым свойством. В частности, оракул O_X умножит «хорошие» базовые состояния на коэффициент (-1) . Это приводит к квантовому состоянию:

$$|\mathcal{U}_1\rangle = O_X |\mathcal{U}\rangle = \frac{1}{\sqrt{n}} \sum_{i=0}^{2^n-1} (-1)^{x_i} |i\rangle.$$

В приведенном выше уравнении $x_i = 1$, если $i \in X$, и $x_i = 0$ в противном случае. Эта операция эквивалентна отражению $|\mathcal{U}\rangle$ через эквивалентную суперпозицию $|\mathcal{B}\rangle$ всех «плохих» состояний: $O_X = 2|\mathcal{B}\rangle\langle\mathcal{B}| - I$.

Далее применяется новая операция рефлексии $H^{\otimes n} R_0 H^{\otimes n} = 2|\mathcal{U}\rangle\langle\mathcal{U}| - I$ к полученному новому состоянию $O_X |\mathcal{U}\rangle$. Здесь R_0 помечает все состояния, кроме состояния $|0\rangle$, с коэффициентом (-1) и в сочетании с операцией $H^{\otimes n}$, которая преобразует состояние $|0\rangle$ в $|\mathcal{U}\rangle$; можно видеть, что общая операция является отражением через состояние $|\mathcal{U}\rangle$. Оба отражения вместе называются итерацией Гровера, и после одной такой итерации мы переходим в состояние $|\mathcal{U}_2\rangle = \sin(2\theta + \theta)|\mathcal{G}\rangle + \cos(2\theta + \theta)|\mathcal{B}\rangle$.

Примечание. Комбинация двух отражений, сначала вокруг «плохих» состояний $|\mathcal{B}\rangle$ и, наконец, вокруг начального состояния $|\mathcal{U}\rangle$, приводит к общему эффекту отклонения от начального состояния $|\mathcal{U}\rangle$ на угол 2θ , так что результирующее состояние становится ближе к «хорошим» состояниям. То, что сочетание двух отражений приводит к такому вращению, можно увидеть геометрически из рис. 23.

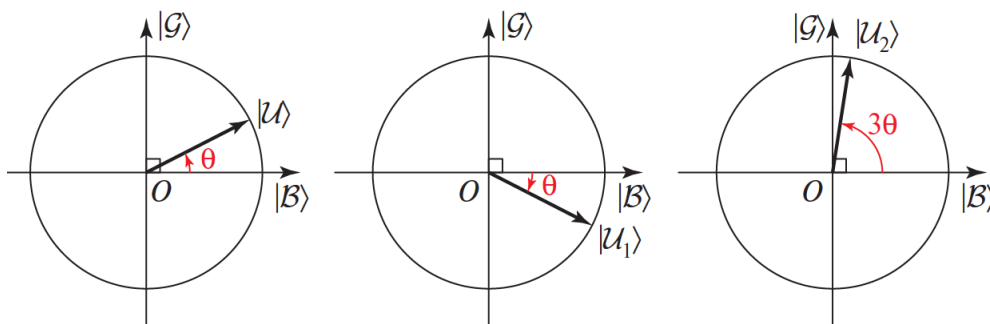


Рис.23. Визуальное представление состояния $|\mathcal{U}\rangle$ и как геометрически отражается итерацией Гровера для приближения к состоянию $|\mathcal{G}\rangle$

Особенностью алгоритма Гровера является то, что квантовое состояние n кубитов после каждой полной итерации Гровера может быть записано как линейная комбинация двух состояний $|\mathcal{G}\rangle$ и $|\mathcal{B}\rangle$ с вещественными коэффициентами,

Следовательно, это квантовое состояние может быть представлено в виде единичного вектора в двумерном пространстве, причем оси x и y представляют $|\mathcal{B}\rangle$ и $|\mathcal{G}\rangle$ соответственно. Как следствие, алгоритм Гровера можно представить геометрически в виде «направленной диаграммы со стрелками». Исходное состояние $|\mathcal{U}\rangle$ имеет угол θ с осью $|\mathcal{B}\rangle$, как показано ранее в виде $|\mathcal{U}\rangle = \sin(\theta)|\mathcal{G}\rangle + \cos(\theta)|\mathcal{B}\rangle$. Следовательно, единичный вектор, представляющий это начальное состояние $|\mathcal{U}\rangle$, почти выровнен с осью $|\mathcal{B}\rangle$, т. е. угол θ мал, как размерность пространства решений ‘хороших’ состояний намного меньше, чем 2^n , как показано на рис.23 (а). Первая итерация по методу Гровера соответствует повороту единичного вектора $|\mathcal{U}\rangle$ на угол 2θ к оси $|\mathcal{G}\rangle$ относительно единичного вектора $|\mathcal{U}_1\rangle$, как указано в уравнении $|\mathcal{U}_2\rangle = \sin(2\theta + \theta)|\mathcal{G}\rangle + \cos(2\theta + \theta)|\mathcal{B}\rangle$. Следовательно, квантовое состояние n кубитов перемещается ближе к оси $|\mathcal{G}\rangle$, как показано на рис.23 (б).

Преимущество геометрической визуализации заключается в том, что она представляет математические формулы, приведенные ранее. В частности, визуализация показывает, что итерации Гровера приближают вектор $|\mathcal{U}\rangle$ к желаемому решению $|\mathcal{G}\rangle$ за конечное число шагов.

Примечание. Недостаток этой визуализации в том, что она скрывает, как правило, значительно отличающуюся размерность пространства решений по сравнению с полным гильбертовым пространством. В частности, диаграмма со стрелками представляет пространство решений в виде $1D$ -подпространства в $2D$ -пространстве. Следовательно, суперпозиция представлена абстрактным образом с единичным вектором, имеющим компоненты в $|\mathcal{G}\rangle$ и $|\mathcal{B}\rangle$, но тот факт, что начальное состояние является однородной суперпозицией всех состояний, представлен слабо. «Диаграмма со стрелками» дает точное представление об итерационной математике Гровера, но понятие интерференции представлено нечетко. На самом деле, вращение стрелки, по-видимому, скорее подчеркивает классический механизм, подобный повороту ручки. Наконец, измерение можно представить в виде проекции на оси x и y . Однако этому опять же препятствует неполное представление размерности.

Применяя эту итерацию Гровера k раз, приходим к состоянию:

$$|\mathcal{U}_k\rangle = \sin((2k+1)\theta)|\mathcal{G}\rangle + \cos((2k+1)\theta)|\mathcal{B}\rangle.$$

Используя это выражение, можно видеть, что если выберем k как $\frac{\pi}{4\theta} - \frac{1}{2}$, то получим $\sin((2k+1)\theta) = 1$ и $|\mathcal{U}_k\rangle = |\mathcal{G}\rangle$. Этот выбор возможен, если значение $\frac{\pi}{4\theta} - \frac{1}{2}$ является целым числом; если это не так, то принимаем k близким к этому значению и получаем $|\mathcal{U}_k\rangle \approx |\mathcal{G}\rangle$. Идея заключается в том, что это вращение выполняется k раз таким образом, чтобы n кубитов находились в состоянии, очень близком или равном состоянию $|\mathcal{G}\rangle$, которое представляет собой равномерную суперпозицию «хороших» состояний. В обоих случаях получаем состояние, которое очень близко или даже равно суперпозиции ‘хороших’ состояний.

Возвращаясь к лабиринту: алгоритм Гровера на шаге 2 приводит состояние n кубитов к такому состоянию, при котором большая часть веса приходится на «хорошие» пути, ведущие к выходу. Более того, операции, описанные на шагах 1 и 2, могут быть выполнены за заданный промежуток времени.

Шаг 3: Измерение. Наконец, измеряем состояние n кубитов. После шага 2 состояние n кубитов равно $|\mathcal{U}_k\rangle$, что равно или очень близко к суперпозиции $|\mathcal{G}\rangle$ по всем возможным 'хорошим' базисным состояниям. Затем, после измерения, вероятность нахождения одного из «хороших» базисных состояний, представляющих элемент с желаемым свойством, равна или почти равна единице. Более того, в квантовой механике состояние системы после измерения возвращается к базовому состоянию, которое соответствует найденному результату. Следовательно, после измерения кубиты с вероятностью 1 (или почти 1) находятся в "хорошем" состоянии, то есть в базовом состоянии, которое представляет собой решение задачи поиска.

Что касается лабиринта: с помощью измерений на этом последнем шаге находим один путь. Чем больше вес на пути, тем больше вероятность, что выберем этот путь. А алгоритм Гровера гарантирует, что «хорошие» пути, ведущие к выходу, в совокупности имеют вес, равный почти 1. Таким образом, вероятность того, что найдем такой успешный путь, становится очень большой.

Д. Особенности трех ключевых квантовых принципов в алгоритме Гровера

Каждый из трех этапов алгоритма Гровера использует один из ключевых квантовых принципов: суперпозицию, интерференцию и измерение. Следовательно, объяснение алгоритма - хороший способ познакомиться со средствами вычислений, которые предлагает квантовая механика.

Шаг 1 алгоритма Гровера вводит суперпозицию в качестве ключевого квантового принципа: начальное состояние квантовых битов в квантовом компьютере преобразуется в линейную комбинацию всех базисных состояний, где некоторое базисное состояние представляет собой возможное решение в рамках поиска и имеет определенную амплитуду. Затем кубиты могут находиться в состояниях, которые являются суперпозициями других состояний.

Шаг 2 знакомит с ключевым принципом интерференции. Квантовое состояние кубитов эволюционирует таким образом, что амплитуды базисных состояний, представляющих «плохие» решения задачи поиска, становятся близкими к нулю, а амплитуды «хороших» базисных состояний растут. В результате, состояния кубитов могут влиять друг на друга, исчезая или становясь более явными.

На шаге 3 алгоритма Гровера представлен ключевой квантовый принцип измерения. Состояние кубитов с вероятностью, близкой к 1, возвращается к базовому состоянию, представляющему решение в рамках поиска. Следовательно, при измерениях состояния кубитов совершают переходы от суперпозиций базисных состояний к единичным базисным состояниям [15].

Алгоритм Гровера является одним из основных алгоритмов, предлагаемых в качестве доказательства того, что квантовые компьютеры могут обеспечить преимущество перед классическими компьютерами. Это включает в себя «oracle» (внешнюю квантовую подпрограмму), которая должна быть указана для данного приложения и внутренняя структура которой не является частью формального масштабирования квантового ускорения, гарантируемого алгоритмом. Алгоритм Гровера также требует экспоненциально большого количества шагов для достижения успеха, что ставит вопрос о его реализации на быстродействующем оборудовании без ошибок и даже на квантовых компьютерах с исправленными ошибками. Опишем квантовый алгоритм, реализуемый на классическом компьютере, который выполняет задачу Гровера за линейное число обращений к оракулу (экспоненциально меньшее число, чем алгоритм Гровера) и демонстрируем этот алгоритм в явном виде для задач булевой выполнимости (3-SAT). Подразумевается, что априорного теоретического квантового ускорения, связанного с алгоритмом Гровера, не существует. Рассматриваем возможность практического ускорения, возможность, которая зависит от природы квантовой схемы, связанной с оракулом. Неблагоприятное масштабирование вероятности успеха алгоритма Гровера, которое при наличии шума уменьшается по экспоненте от экспоненты числа кубитов, делает практическое ускорение нереалистичным даже при чрезвычайно оптимистичных предположениях как о качестве оборудования, так и о доступности.

Среди возможных применений квантовых вычислений доминируют два класса алгоритмов. Первый класс вычисляет нетривиальный результат, а затем извлекает его с помощью квантового преобразования Фурье. К этому классу относятся оригинальный алгоритм Шора для целочисленной факторизации, а также алгоритм оценки квантовой фазы, предложенный для решения задач квантовой химии, и несколько других алгоритмов. Некоторые из этих алгоритмов, в частности алгоритм Шора, обеспечивают экспоненциальное ускорение по сравнению с любыми известными классическими методами, хотя и только для нескольких довольно специфических приложений.

Ко второму классу относится алгоритм Гровера (*GrA*) и его обобщения, такие как амплитудное усиление. Алгоритм Гровера обещает менее впечатляющее квадратичное ускорение, но, в свою очередь, пользуется широкой популярностью благодаря множеству возможных вариантов использования. Теоретически, решение довольно большого числа задач можно было бы ускорить, просто переставив критическую часть классического алгоритма с помощью вызова процедуры Гровера, реализованной на квантовом компьютере. Также привлекательно, что квадратичное ускорение алгоритма Гровера может быть основано на твердых математических основаниях и является доказуемо оптимальным при определенных предположениях, в отличие от алгоритма Шора, где ускорение является только предположительным. Это квадратичное ускорение очень удобно в теоретическом плане, поскольку не требует каких-либо знаний оракула, который кодирует проблему в квантовом алгоритме. Класс задач, для решения которых может быть применен алгоритм Гровера, включает в себя *NP*-полные задачи, которые являются чрезвычайно сложными в вычислительном отношении.

Алгоритм Гровера решает проблему инвертирования неизвестной функции. Учитывая функцию $y = f(b)$, где b может принимать $N = 2^n$ значений, алгоритм Гровера находит значение $b = f^{-1}(y)$ для данного y всего за \sqrt{N} вызовов функции квантовый оракул (*quantum oracle*), которая реализует $f(b)$, в то время как для наивного исчерпывающего поиска потребовалось бы около N вызовов соответствующей классической функции.

Теоретическое ускорение, предложенное Гровером, имеет особую природу: это абстрактное ускорение, которое рассматривает оракул как функцию черного ящика и рассчитывает вычислительные затраты исключительно с точки зрения количества обращений к оракулу. Однако при любой реальной реализации оракул должен быть реализован как конкретная квантовая схема, следовательно, внутренняя структура $f(b)$ должна быть раскрыта. В частности, вычислительные затраты на один вызов оракула $f(b)$ могут (почти всегда будут) каким-то образом зависеть от N .

Та же схема «оракула», которая определяет реализацию алгоритма Гровера на квантовом компьютере, используется в классическом симуляторе, который ограничен только запутанностью (тензорной сетью). Если этот симулятор может вычислить одну схему оракула со сложностью выше, чем $\sqrt{2^n}$, квантово-инспирированный алгоритм Гровера (*QiGA*) решает задачу алгоритма Гровера параметрически быстрее, чем это сделал бы квантовый компьютер. Даже когда симулятор масштабируется как менее благоприятная экспонента, наглядно демонстрируем случаи, когда моделирование оракула может быть выполнено всего за несколько часов на настольном компьютере с $n \approx 40$ кубитами и, вероятно, на суперкомпьютере с $n \approx 80$ кубитами.

Тем не менее, для решения таких задач алгоритмом Гровера потребовалось бы по меньшей мере $\sqrt{2^{80}} = 2^{40} \approx 10^{12}$ применений схемы оракул на квантовом компьютере, что приводит к астрономически большому времени на решение даже при благоприятных предположениях об аппаратном обеспечении.

В основе лежит наблюдение о том, что в промежутках между обращениями к оракулу уровень запутанности, присутствующий во внутреннем состоянии квантового компьютера, работающего под управлением алгоритма Гровера, чрезвычайно низок. В промежутке между каждой итерацией алгоритма Гровера энтропия запутанности между любыми двумя подгруппами кубитов составляет не более $\log(2)$. Другими словами, алгоритм Гровера сильно использует способность квантовой механики создавать суперпозиции (квантовый параллелизм), но практически не использует возможность запутывать состояния.

На рис. 24 показана энтропия запутанности квантового состояния при моделировании алгоритма Гровера для $n = 10$ кубитов с использованием квантовой схемы, описанной ниже, для реализации операторов оракула и диффузии.

Промежуточные этапы между каждой полной итерацией показывают свойства состояния после каждого уровня квантовых схем, которые реализуют операторы оракул и диффузии (*diffusion*).

В первой половине алгоритма энтропия определяется шагом оператора оракул, а во второй половине алгоритма - шагом оператора диффузии. Во время моделирования ранг состояния матричного произведения (*MPS – matrix product state*) (не показан) увеличивается и уменьшается в виде пилообразной формы с максимумами $\chi = 11$ в течение итерации и минимумами $\chi = 2$ между каждой итерацией.

Пунктирная линия вдоль нижней огибающей - теоретическое предсказание запутанности состояний между итерациями алгоритма (колебания - небольшой эффект $1/N$). На нижней панели показан в рамке на верхней панели масштаб области, обозначающей подэтапы, на которых работают схемы оракула и диффузии.

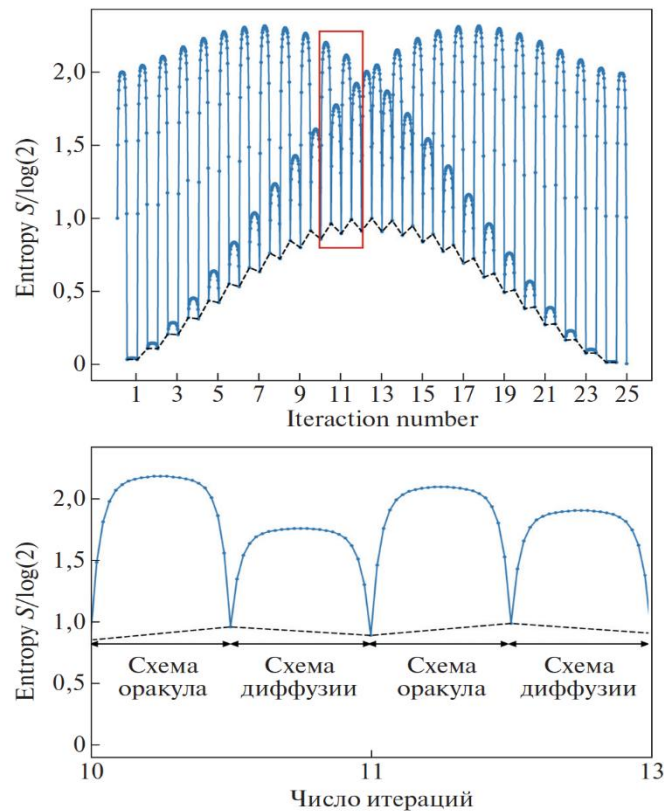


Рис.24. Энтропия запутанности квантового состояния в единицах $\log(2)$ при моделировании алгоритма Гровера для $n = 10$ кубитов при оптимальном числе $r = 25$ итераций [16]

В первой половине алгоритма энтропия определяется шагом оператора оракул, а во второй половине алгоритма – шагом оператора диффузии. Во время моделирования ранг состояния матричного произведения (*MPS – matrix product state*) (не показан) увеличивается и уменьшается в виде пилообразной формы с максимумами $\chi = 11$ в течение итерации и минимумами $\chi = 2$ между каждой итерацией. Пунктирная линия вдоль нижней огибающей - теоретическое предсказание запутанности состояний между итерациями алгоритма (колебания – небольшой эффект $1/N$). На нижней панели показан в рамке на верхней панели масштаб области, обозначающей подэтапы, на которых работают схемы оракула и диффузии.

Как утверждается, запутанность между итерациями никогда не превышает $\log(2)$, значение, которое достигается, когда алгоритм выполняется наполовину. Энтропия запутанности действительно повышается при частичном применении схем операторов оракул и диффузии. Значение этого барьера запутанности внутри оператора оракул сильно зависит от проблемы и будет определять производительность алгоритма Гровера, основанного на квантовой технологии.

Пример: квантовые схемы для операторов оракул и диффузии. Для получения графика энтропии запутанности на рис.25, который показывает запутанность не только между итерациями Гровера, но и внутри под-этапов схем оракул и диффузии, использовали схемы, показанные на рис. 25 ниже.

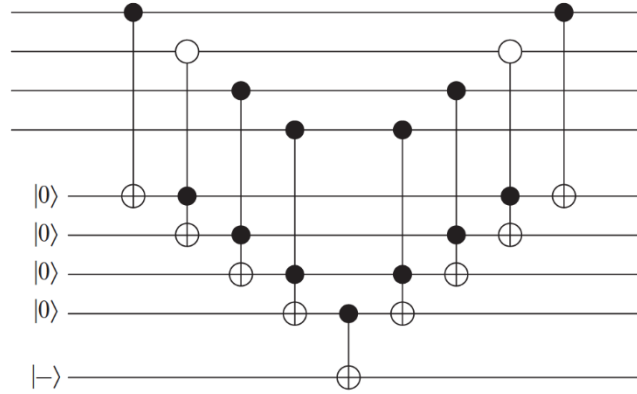


Рис. 25. Схема моделирования оператора оракул

Эти схемы предназначены для случая, когда известна целевая битовая строка w . (Цель для оператора разделения всегда известна, поскольку он может быть реализован как оракул, который нацелен на $|000\dots 0\rangle$, предварительно и пост-обработанный логическим элементом Адамара на каждом кубите.) В приведенной ниже схеме используются не более чем трех-кубитные вентили. Это контрастирует с реализацией, которую иногда можно увидеть, когда оракул реализуется с помощью «многоуправляемого» шлюза, что эквивалентно нашему наблюдению в основном тексте о том, что оракул в принципе всегда может быть реализован с помощью $\text{rank } \chi = 2$ МРО (для случая единственной целевой битовой строки w). Для конкретного случая с четырьмя кубитами и целевой битовой строкой $w = 1011$ схема моделирования оракула (см. рис. 3), используемая на рис. 2, была в виде:

Для алгоритма Гровера на n кубитах описанный выше оператор инициализируется путем подготовки $n + 1$ дополнительных вспомогательных кубитов, n в состоянии $|0\rangle$ и $n + 1$ кубита в состоянии $|-\rangle = H|1\rangle$. При использовании гейтов Тоффоли, воздействующих на верхний и нижний регистры, вспомогательные элементы меняются, указывая, что каждый кубит целевой битовой последовательности найден (верхний контроль) и что все предыдущие биты были найдены (нижний контроль). Если это так, то следующий вспомогательный кубит получает значение 1. Путем непосредственной проверки легко увидеть, что применение приведенной выше схемы к $|b\rangle|0000-\rangle_A$ приводит к $\pm|b\rangle|0000-\rangle_A$ в зависимости от того, $b = w$ или нет.

Важное наблюдение заключается в том, что возможности вычислить влияние схемы оракула на начальное состояние (состояние после оракула) всегда достаточно для завершения алгоритма Гровера за один вызов оракула. На самом деле, достаточно даже просто вычислить $\sim \log(N)$ амплитуды этого состояния в базисе $|\pm\rangle$. Таким образом, квантовый алгоритм отличается от простого моделирования алгоритма Гровера, которое все равно потребовало бы экспоненциально большого числа $\sqrt{2^n}$ приложений схемы оракул.

Пример: об уровне запутанности внутри алгоритма Гровера. Тип классического вычисления, который рассмотрим, включает в себя представление квантового состояния в виде тензорной сети, в частности состояния матричного произведения (MPS). MPS сжимает квантовое состояние, преобразуя его в сеть меньших тензоров, сжатых в одномерную структуру, подобную цепи. Для состояний с низкой или умеренной вовлеченностью ранг MPS или размерность индексов «связи», соединяющих тензоры MPS, могут быть выбраны относительно малыми, но при этом представлять состояние с очень высокой или даже идеальной точностью. В контексте алгоритма Гровера можно обнаружить, что после любого применения U_w или U_s внутреннее состояние $|\psi\rangle$ квантового компьютера находится в суперпозиции $|s\rangle$ и $|w\rangle$, $|\psi\rangle = \alpha|s\rangle + \beta|w\rangle$ и $|\alpha|^2 + |\beta|^2 = 1$, т.е. в суперпозиции двух не расцепленных состояний ($1 + S$ состояний в общем случае). Отсюда следует, что $|\psi\rangle$ может быть приведено к виду,

$$|\psi\rangle = [\alpha \quad \beta] \begin{bmatrix} |w_1\rangle & 0 \\ 0 & |+\rangle \end{bmatrix} \begin{bmatrix} |w_2\rangle & 0 \\ 0 & |+\rangle \end{bmatrix} \dots \begin{bmatrix} |w_{n-1}\rangle & 0 \\ 0 & |+\rangle \end{bmatrix} \begin{bmatrix} |w_n\rangle & 0 \\ 0 & |+\rangle \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Такое состояние минимально запутано и может быть легко сохранено в памяти классического компьютера с линейными затратами в количестве кубитов. Другими словами, хотя алгоритм Гровера использует преимущества квантового параллелизма (т.е. суперпозиции), в большей части алгоритма он использует очень малую степень запутанности. Единственным возможным исключением является то, что схема оракула была применена лишь частично.

Чтобы использовать алгоритм Гровера в задаче классификации, необходимо переформулировать задачу классификации как задачу поиска, заменив оракул Гровера вариационным алгоритмом и добавив квантовые элементы (*AND*, *XOR* и *OR*). Чтобы ввести понятие классификации в алгоритм Гровера, используется метод ядра, который является методом, более адаптированным к сложным задачам; кроме того, другие исследовательские работы пытаются решить задачу классификации, применяя алгоритм Гровера в различных методах.

Новый метод может быть введен на основе добавления схемы Гровера после вариационного алгоритма. Как известно, можно применить схему Гровера для усиления амплитуды, поэтому в предлагаемом методе используют ее для ускорения процесса обучения за счет усиления амплитуды целевого состояния. Это упрощает идентификацию на этапе измерения, а также увеличивает амплитуду правильного выходного состояния в суперпозиции всех возможных выходных состояний и обновляет веса для получения более точной модели.

Схема почти такая же, как и в вариационной схеме. Однако получили классификацию, измерив вероятность расположения кубитов в определенной конфигурации, и несколько раз выполнили ту же схему, что и выше, чтобы получить значения функции потерь. Было подсчитано, сколько раз конфигурация появлялась в классифицированном наборе данных. Он двумерный, потому что используем два кубита; далее присваиваем значения параметрам схемы, а затем изменяем масштаб этих входных значений, чтобы они соответствовали углам. На рис. 26 показана квантовая схема, применяемая в этом методе, с использованием тех же шагов, что и ранее, затем добавляется схема Гровера с использованием элементов Адамара, *CNOT*, управляемых *Z*-элементов и *X*-элементов Паули.

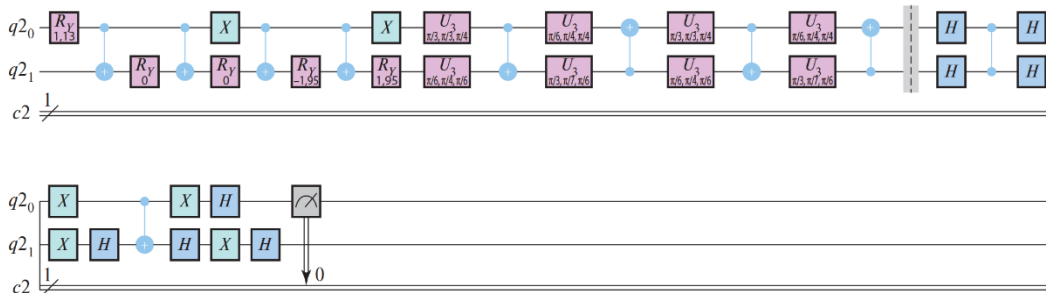


Рис. 26. Схема квантового вариационного перцептрона с встроенным алгоритмом Гровера

Была рассчитана точность для каждой итерации модели: квантового вариационного перцептрона с помощью *Grover* и вариационного перцептрона с использованием *ibmq qasm simulator*, который является одним из квантовых симуляторов *IBM*. Этот *QASM*-симулятор позволяет генерировать квантовые схемы как в идеальном случае, так и с учетом моделирования помех, используя максимум 32 кубита. Результаты проверяют, соответствует ли предложенная модель наборам для обучения и проверки, используя результат работы вариационного перцептрона в качестве эталона и сравнивая обе квантовые модели с классической моделью, чтобы показать квантовое преимущество.

В табл. 4 показано, как суммировали результаты трех различных моделей, и сравнение точности модели на обучающем наборе с поверочным набором может дать представление о способности модели обобщаться на новые данные.

Табл. 4. Оценки точности обучения и верификации классического перцептрона (CP), квантового вариационного перцептрона (QVP) и квантового вариационного перцептрона с встроенным алгоритмом Гровера (GVP-G)

Iter	CP		QVP		QVP-Grover	
	Acc train	Acc valid	Acc train	Acc valid	Acc train	Acc valid
1	0,49	0,81	0,45	0,36	0,47	0,40
10	0,91	0,92	0,53	0,36	0,47	0,40
15	0,93	0,93	0,53	0,34	0,49	0,45
20	0,94	0,94	0,53	0,32	0,50	0,48
25	0,94	0,94	0,26	0,40	0,55	0,50
30	0,95	0,94	0,29	0,46	0,56	0,53
35	0,95	0,95	0,62	0,50	0,59	0,58
40	0,95	0,95	0,68	0,66	0,64	0,60
45	0,96	0,95	0,73	0,66	0,77	0,73
50	0,96	0,95	0,84	0,66	0,85	0,80
55	0,96	0,96	0,99	0,66	0,91	0,90
60	0,96	0,96	0,94	0,60	0,97	0,91
65	0,96	0,96	0,96	0,70	1,00	0,91
70	0,96	0,96	0,96	0,74	1,00	0,94
75	0,97	0,96	0,98	0,81	1,00	0,97
80	0,97	0,96	1,00	0,82	1,00	0,99
85	0,97	0,96	1,00	0,82	1,00	0,99
90	0,97	0,96	1,00	0,84	1,00	0,99
95	0,97	0,96	1,00	0,90	1,00	0,99
100	0,97	0,96	1,00	0,90	1,00	0,99

В целом, точность модели на обучающей выборке будет выше, чем ее точность на проверочном наборе, потому что модель «видела» обучающие данные во время обучения и научилась правильно их классифицировать.

Соответственно, можем предположить, что ни одна из трех моделей не подходит полностью. Разница в том, что, как правило, когда модель достигает высокой точности проверки за меньшее количество итераций, ее можно считать более эффективной, поскольку для получения точных прогнозов требуется меньше точек данных. Квантово-вариационная модель персептрона и квантово-вариационный персептрон с алгоритмом Гровера имеют высокую точность проверки - 99% и достигают этой точности после нескольких итераций, что указывает на то, что они могут быть более эффективными, чем классическая модель персептрона. Классическая модель персептрона имеет самую низкую точность проверки - 96%, но она достигла этой точности после большого количества итераций, что показывает, что обе квантовые модели лучше. Однако она может быть более надежной и применимой к новым данным.

Важно отметить, что одного количества итераций недостаточно, чтобы решить, какая модель лучше. Мы можем объяснить это использованием квантовых компьютеров, которые, согласно законам квантовых вычислений, имеют более мощные и эффективные способы прогнозирования результатов. Кроме того, благодаря использованию схемы Гровера предложенная модель получила конфигурацию, позволяющую более плавно и быстро измерять вероятность, что позволило быстро получить ожидаемое значение вероятности и весовые коэффициенты для достижения высокой точности.

Величина потерь показывает, насколько плохо или хорошо модель ведет себя после каждой итерации оптимизации, поэтому попытались построить график функции потерь для трех методов: классического восприятия, квантово-вариационного персептрона и квантово-вариационного персептрона с Гровером, на рис. 27 показано, что функция потерь хорошо сходится в течение 100 итераций по сравнению с другими моделями (CP и QVP).

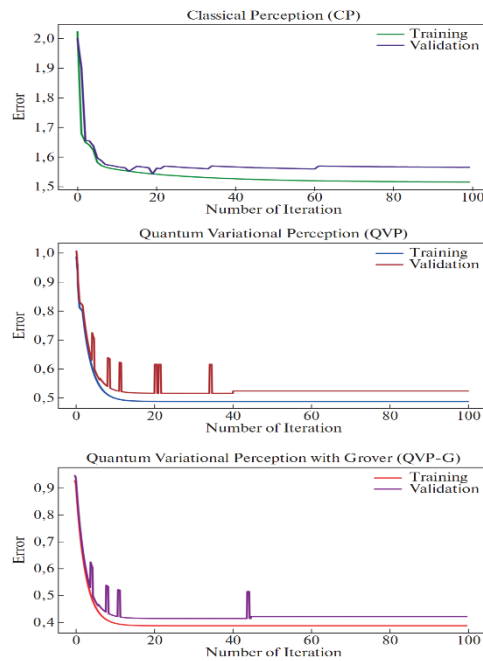


Рис.27. Функции потерь алгоритмов CP, QVP, and QVP – G [17]

Иногда крайне важно остановить обучение, когда модель начинает перегружаться; это можно сделать, отслеживая потери или точность с течением времени, поэтому следующим шагом является сравнение функции потерь. Используем функцию потерь для оптимизации квантовой или классической модели машинного обучения, измеряя разницу между прогнозируемыми результатами модели и фактическими результатами. Обычно количественно оцениваем эту разницу как ошибку, в точности равную сумме ошибок, допущенных для каждого примера в наборе данных; именно поэтому обучаем модель минимизировать эту ошибку.

Как известно, меньшее значение функции потерь обычно указывает на лучшую производительность, поскольку это означает, что предсказания модели ближе к фактическим значениям. Таким образом, функция потерь *QVP-G* (0.4) будет считаться наиболее эффективной, поэтому можем предположить, что предложенный метод улучшает производительность модели.

Следовательно, использование алгоритма Гровера в квантовом машинном обучении, в частности в модели перцептрона, является многообещающим, поскольку этот подход позволяет сократить количество итераций, необходимых для достижения наилучшей

Таким образом, квантовые нейронные сети (*QNN*) – подкласс вариационных квантовых алгоритмов, состоящий из квантовых схем, которые содержат параметризованные логические операции. Информация сначала кодируется в квантовое состояние с помощью процедуры подготовки состояния или отображения свойств. Выбор отображения свойств обычно направлен на повышение производительности квантовой модели и, как правило, не оптимизируется и не обучается. Как только данные кодируются в квантовое состояние, применяется вариационная модель, содержащая параметризованные элементы управления, которая оптимизируется для конкретной задачи. Это происходит за счет минимизации функции потерь, когда выходные данные квантовой модели могут быть извлечены из классической функции пост-обработки, которая применяется к результату измерения.

В [18, 19] рассмотрены некоторые примеры и особенности применения квантового глубокого обучения в интеллектуальной когнитивной робототехнике.

Выводы

Рассмотрены дополнительные особенности моделей квантового перцептрона и модели инженерии квантового глубокого обучения при применении инструментария оптимизации баз знаний интеллектуальных регуляторов на основе *QCOpiKBTM* в задачах квантового интеллектуального управления роботизированными социотехническими системами в проектах «Индустрия 4.0 / 5.0 / 6.0».

За счет операторов квантовых вычислений классические данные кодируются кубитами, осуществляется оптимальный выбор квантовой корреляции между искомыми решениями и применением конструктивной интерференции извлекается искомый результат путем измерения. Таким образом, разработанный интеллектуальный инструментарий *SCOptKBTM* и *QCOptKBTM* включает принципы квантового глубокого обучения, и также как в случае технологии мягких вычислений, формирует оптимальную структуру теперь квантовой нейронной сети, а за счет применения квантового генетического алгоритма ускоряет поиск искомого решения.

Рассмотрены прикладные модели квантового глубокого обучения для применения в проекте «Индустрия 4.0 / 5.0».

Список источников

1. Physical limits and information bounds of micro control. Part 2: Quantum soft computing and quantum searching algorithms / S. V. Ulyanov [et al.] // Proceedings of the 1998 International Symposium on Micromechatronics and Human Scienc (MHS'98). Nagoya, Japan. – 1998. – Pp. 217-224. – DOI: 10.1109/MHS.1998.745785.
2. Information analysis of quantum gates for simulation of quantum algorithms on classical computers / S. V. Ulyanov [et al.] // International Conference on Quantum Communication, Measurements and Computing (QCM&C'2000). – Capri, Italy, 2000. KluwerAcad. PlenumPubl. – 2001. – pp. 207-214.
3. Ulyanov S. V., Litvintseva L. V., Hagiwara T. Design of self-organized intelligent control systems based on quantum fuzzy inference: intelligent system of systems engineering approach // IEEE International Conference on System, Man and Cybernetics (SMC'2005). – Hawaii, USA, 2005. – Vol. 4. – Pp. 3835-3840. – DOI: 10.1109/ICSMC.2005.1571744.
4. Valdez F., Melin P. A review on quantum computing and deep learning algorithms and their applications // Soft Computing. – Springer-Verlag GmbH Germany, part of Springer Nature, 2022. – DOI: 10.1007/s00500-022-07037-4 (published online: 07 April 2022).
5. Ulyanov S.V. Quantum Soft Computing in Control Process Design: Quantum Genetic Algorithms and Quantum Neural Network Approaches // Proceedings World Automation Congress, Fifth Intern. Symposium on Soft Computing for Industry. – Seville, Spain June 28th-July 1st, 2004 (paper No ISSCI028).
6. Quantum machine learning for chemistry and physics / M. Sajjan [et al.] // Chem Soc Rev. – 2022. – Vol. 51. – Pp. 6475-6573. – DOI: 10.1039/d2cs00203e.
7. A Quantum Neural Network-Based Approach to Power Quality Disturbances Detection and Recognition / D.-G. Li [et al.] // arXiv.org e-Print archive. – arXiv:2406.03081v1 [quant-ph] 5 Jun 2024.
8. Huang H-Y., Kueng R., Preskill J. Information-theoretic bounds on quantum advantage in machine learning // arXiv.org e-Print archive. – arXiv:2101.02464v1 [quant-ph] 7 Jan 2021.
9. Synergy of machine learning with quantum computing and communication / Debasmita Bhoumik, Susmita Sur-Kolay, Latesh Kumar K. J., Sundaraja Sitharama Iyengar // arXiv.org e-Print archive. – arXiv:2310.03434v1 [quant-ph] 5 Oct 2023.
10. A General Approach to Dropout in Quantum Neural Networks / F. Scala, A. Ceschini, M. Panella, D. Gerace // arXiv.org e-Print archive. – arXiv:2310.04120v1 [quant-ph] 6 Oct 2023.
11. A Survey on Quantum Machine Learning: Current Trends, Challenges, Opportunities, and the Road Ahead / Kamila Zaman, Alberto Marchisio, Muhammad Abdullah Hanif, Muhammad Shafique // arXiv.org e-Print archive. – arXiv:2310.10315v1 [quant-ph] 16 Oct 2023.
12. Chen Z.-B. Quantum Neural Network and Soft Quantum Computing // arXiv.org e-Print archive. – arXiv:1810.05025v1 [quant-ph] 10 Oct 2018.
13. sQULearn – A Python Library for Quantum Machine Learning / D. A. Kreplin [et al.] // arXiv.org e-Print archive. – arXiv:2311.08990v1 [quant-ph] 15 Nov 2023.

14. An artificial neuron implemented on an actual quantum processor / F. Tacchino, C. Macchiavello, D. Gerace, D. Bajoni // *npj Quantum Information*. – 2019. – Vol. 5. – Article number: 26 (2019). – DOI: <https://doi.org/10.1038/s41534-019-0140-4>.
15. Explaining Grover’s algorithm with a colony of ants: a pedagogical model for making quantum technology comprehensible / Merel A. Schalkers, Kamiel Dankers, Michael Wimmer, Pieter Vermaas // *arXiv.org e-Print archive*. – arXiv:2405.00014v1 [physics.pop-ph] 9 Feb 2024.
16. Stoudenmire E.M., Waintal X. Grover's Algorithm Offers No Quantum Advantage // *arXiv.org e-Print archive*. – arXiv:2303.11317v1 [quant-ph] 20 Mar 2023.
17. Innan N., Bennai M. Simulation of a Variational Quantum Perceptron using Grover’s Algorithm // *arXiv.org e-Print archive*. – arXiv:2305.11040v1 [quant-ph] 18 May 2023.
18. Интеллектуальная когнитивная робототехника. Ч. 4.1. Квантовый «сильный» вычислительный интеллект в интеллектуальном управлении роботизированными автономными системами в «Индустрия 4.0 / 5.0» / Р. Ю. Капков, А.Г. Решетников, О. Ю. Тятюшкина, С. В. Ульянов. – Москва: Курс, 2024.
19. Интеллектуальная когнитивная робототехника. Ч. 4.2. Квантовый «сильный» вычислительный интеллект в интеллектуальном управлении роботизированными социотехническими системами / Р. Ю. Капков, А.Г. Решетников, О. Ю. Тятюшкина, С. В. Ульянов. – Москва: Курс, 2024.