

УДК 004.032.26

СТАТИСТИЧЕСКОЕ ОНЛАЙН-ОБУЧЕНИЕ В РЕКУРРЕНТНЫХ И ПРЯМОГО РАСПРОСТРАНЕНИЯ КВАНТОВЫХ НЕЙРОННЫХ СЕТЯХ

© 2023 г. С. В. Зуев^{1,*}

Представлено академиком РАН А.Л. Семеновым

Поступило 24.08.2023 г.

После доработки 15.09.2023 г.

Принято к публикации 24.10.2023 г.

Цели. Для систем адаптивного искусственного интеллекта вопрос о возможности онлайн-обучения является особенно важным, так как такое обучение и обеспечивает адаптацию. Цель работы – рассмотреть методы квантового машинного онлайн-обучения для наиболее распространенных двух архитектур квантовых нейронных сетей: прямого распространения и рекуррентной.

Методы. В работе используется доступный на PyPI модуль `quantumz` для эмуляции квантовых вычислений и создания искусственных квантовых нейронных сетей. Кроме того, для преобразования размерностей данных используется модуль `genset`, обеспечивающий обратимую трансформацию размерностей без потери информации. Данные для экспериментов взяты из открытых источников. В работе реализуется метод машинного обучения без оптимизации, предложенный автором ранее.

Результаты. Представлены и экспериментально подтверждены алгоритмы онлайн-обучения для рекуррентной и прямого распространения квантовой нейронной сети.

Выводы. Предложенные алгоритмы обучения являются инструментами для интеллектуального анализа данных, а также для создания адаптивных интеллектуальных систем управления. Разработанное алгоритмическое обеспечение может полностью раскрыть свой потенциал только на квантовых вычислителях, но, в случае небольшого числа квантовых регистров, может быть применено и в системах, эмулирующих квантовые вычисления, или в фотонных вычислителях.

Ключевые слова: онлайн-обучение, адаптивный искусственный интеллект, квантовое машинное обучение, квантовая запутанность

DOI: 10.31857/S268695432360129X, EDN: CXDGOO

1. ВВЕДЕНИЕ

Главным возможным преимуществом квантового машинного обучения считается экспоненциальное снижение сложности вычислений. В то время как классический искусственный нейрон может обрабатывать входные данные N измерений, квантовый нейрон может обрабатывать 2^N -мерные данные. Это может значительно ускорить обучение [1]. Тем не менее существующие прототипы квантовых нейронных сетей (см., например, [2–6]) и других машинно обучаемых моделей (работы [7–15]) не показывают настолько высокой эффективности. Это связано с тем, что при проведении квантового вычисления необходимо обеспечить наличие “кубитов, которые могут быть инициализированы произвольными значе-

ниями” [16]. Иными словами, нужно уметь представлять данные в виде квантовых состояний или (что почти то же самое) научиться делать очень точные квантовые гейты.

Представление данных в виде состояния квантовой системы требует умения работать с размерностями данных без потери информации. Потому что размерности данных, представляемых состояниями кубитов, могут принимать только определенные значения, а вырожденные измерения (где установлено какое-то одно значение, скажем, 0) значительно снижают эффективность работы. Далее в настоящем исследовании будет представлен программный модуль `genset`, который доступен на PyPI и успешно справляется с задачей изменения размерности датасета. Надо сказать, что аналитики данных, как правило, прибегают к снижению размерности, для чего используется метод главных компонент, для которого есть и квантовый аналог (QPCA) [17] с заявленным экспоненциальным ускорением. Но бывает и так, что увеличение размерности данных

¹Белгородский государственный технологический университет им. В.Г. Шухова, Белгород, Россия

*E-mail: sergey.zuev@bk.ru

может привести к лучшим результатам анализа и, кроме того, потеря информации (обязательная при применении метода главных компонент) не всегда приемлема.

В работе [18] был предложен метод квантового машинного обучения, основанный на запутанных квантовых состояниях. Отличительной чертой этого метода является то, что при таком обучении не требуется оптимизация и, следовательно, не нужны множественные итерации (эпохи). Была показана работа классификаторов, обученных предложенным способом, на некоторых известных датасетах и проведено сравнение с результатами, полученными другими классификаторами.

В настоящей работе метод квантового машинного обучения, основанный на запутанных состояниях (далее — КМОЗС) будет детально рассмотрен и полностью адаптирован для применения в онлайн-обучаемых моделях квантовых нейронных сетей, используемых для классификации и распознавания.

2. МАТЕРИАЛЫ И МЕТОДЫ

Квантовая запутанность в связи с моделью познания была предложена в 2005 г. [19]. Это модель для семантики комбинаций понятий, выполненная недекомпозиционным способом. В нем рассматриваются возникающие свойства/ассоциации/умозаключения в связи с комбинациями понятий. В КМОЗС эта идея используется для других целей: чтобы обеспечить способ разделения помеченных данных.

Описания квантовых и классических данных в настоящей работе будут сделаны традиционным образом. Множество $\{x^j\}$ наборов из n действительных чисел $x^j = (x_0^j, \dots, x_{n-1}^j)$ с меткой j , определенной для каждого набора, представляет собой *классические* данные. Множество квантовых состояний

$$|q^j\rangle \equiv \sum_{k=0}^{2^n-1} z_k^j |k\rangle \quad (1)$$

с теми же метками j рассматривается как *квантовые* данные. Компоненты z_k^j вектора квантового состояния рассматриваются заданными в определенном вычислительном базисе $|0\rangle, \dots, |2^n-1\rangle$. Эти обозначения являются общепринятыми. Прежде чем будет установлена взаимосвязь между этими данными, опишем структуру пространства состояний квантовой системы.

2.1. Представление классических данных квантовыми данными

Пусть система состоит из n частиц, каждая из которых дает при измерении один из двух результатов — это система из n *кубитов*. Известно, что квантовые состояния описываются векторами с абсолютным значением, равным 1, и векторы, отличающиеся только фазовым коэффициентом $e^{i\phi}$, описывают одно и то же состояние. То есть, для координат $(z_0, \dots, z_{N-1})$ любого экземпляра квантовых данных, можно положить

$$z_0^2 + |z_1|^2 \dots + |z_{N-1}|^2 = 1, \quad (2)$$

где $z_0 \in [0, 1]$, $z_1, \dots, z_{N-1} \in \mathbb{C}$. Это уравнение сферы в пространстве $\mathbb{R}^{2N-1}$, т.е., сферы S^{2N-2} . Каждую ее точку z можно записать в следующих обобщенных сферических координатах:

$$\begin{aligned} z_0 &= \cos \delta_0 \cos \delta_1 \dots \cos \delta_{N-2}, \\ z_1 &= \sin \delta_0 \cos \delta_1 \dots \cos \delta_{N-2} e^{i\gamma_0}, \\ z_2 &= \sin \delta_1 \cos \delta_2 \dots \cos \delta_{N-2} e^{i\gamma_1}, \\ &\dots \\ z_{N-2} &= \sin \delta_{N-3} \cos \delta_{N-2} e^{i\gamma_{N-3}}, \\ z_{N-1} &= \sin \delta_{N-2} e^{i\gamma_{N-2}}, \end{aligned} \quad (3)$$

где $\delta_s \in [0, \frac{\pi}{2}]$, $\gamma_s \in [0, 2\pi)$, $\forall s$.

Пусть имеется классический набор данных x^j , $j = 0, \dots, J-1$, размерности M такой, что значения каждого признака являются целыми числами, лежащими в интервале от 0 до X_m ($m = 0, \dots, M-1, X_m \in \mathbb{N}$). Иными словами,

$$x_m^j \in [0, X_m], \quad x_m^j \in \mathbb{Z}.$$

Даже если исходный набор данных не таков, можно произвести соответствующее преобразование и получить требуемый набор. Такой датасет можно закодировать в набор квантовых состояний (квантовый датасет), если $M = 2 \cdot 2^n - 2$ для какого-то натурального n . Если $N = 2^n$, то квантовый датасет будет определен формулой (1), в которой для каждого j -го экземпляра данных числа z_k^j определены формулой (3), в которой

$$\begin{aligned} \delta_s^j &= \frac{x_s^j}{X_s} \cdot \frac{\pi}{2}, \quad s = 0, \dots, N-2, \\ \gamma_t^j &= \frac{x_t^j}{X_t + 1} \cdot 2\pi, \quad t = N-1, \dots, 2N-3. \end{aligned}$$

Если в исходном датасете $M \neq 2 \cdot 2^n - 2$, то требуется изменить размерность датасета. Это мож-

но сделать с помощью Python модуля genser, алгоритм работы которого описан далее.

2.2. Алгоритм обобщенной сериализации для изменения размерности датасета

В основе изменения размерности датасета без потери информации лежит хорошо известная сериализация, обобщенная следующей теоремой.

Теорема 1. Пусть $R_X^s = \{(x_0, \dots, x_{s-1}), x_j \in [0, X_j], x_j \in \mathbb{Z}, j = 0, \dots, s-1\}$. Множества R_X^s и R_Y^t изоморфны, если

$$\prod_{j=0}^{s-1} (X_j + 1) = \prod_{k=0}^{t-1} (Y_k + 1).$$

Доказательство. Предъявим биекцию в явном виде. Заметим, что если выполнено условие теоремы, то оба произведения (равные одному и тому же числу) имеют единственное разложение на простые сомножители (основная теорема арифметики). Имеем следующие альтернативы:

- а) все $X_j + 1$ простые и все $Y_k + 1$ тоже простые;
- б) все $X_j + 1$ простые, а среди $Y_k + 1$ есть составные;
- с) среди $X_j + 1$ есть составные, а все $Y_k + 1$ простые;
- д) среди $X_j + 1$ есть составные, и среди $Y_k + 1$ есть составные.

В случае а), очевидно, $s = t$ и можно построить биекцию вида $y_k = x_j$ для $X_j = Y_k$.

Если имеет место случай б), то, очевидно, $t < s$, и среди $Y_k + 1$ найдутся такие, которые являются произведением нескольких сомножителей вида $X_j + 1$:

$$Y_k + 1 = (X_{j_1} + 1) \cdot \dots \cdot (X_{j_r} + 1).$$

Для такого случая положим

$$y_k = (X_{j_1} + 1) \cdot \dots \cdot (X_{j_{r-1}} + 1) x_{j_r} + (X_{j_1} + 1) \cdot \dots \cdot (X_{j_{r-2}} + 1) x_{j_{r-1}} + \dots + (X_{j_1} + 1) x_{j_2} + x_{j_1}$$

в качестве прямого отображения и

$$x_{j_1} = y_k \bmod (X_{j_1} + 1),$$

$$x_{j_2} = y_k \setminus (X_{j_1} + 1),$$

$$x_{j_3} = y_k \setminus ((X_{j_1} + 1)(X_{j_2} + 1)),$$

...

$$x_{j_r} = y_k \setminus ((X_{j_1} + 1) \dots (X_{j_{r-1}} + 1))$$

в качестве обратного. Легко проверить, что это взаимно-однозначное отображение. Для остав-

шихся простыми сомножителей поступаем так же, как в случае а).

При выполнении с), биекция строится аналогично случаю б) с той лишь разницей, что теперь $s < t$ и X нужно поменять местами с Y .

В случае d) нужно заметить, что между R_X^s и R_Y^t всегда можно поставить множество $R_Z^p = \{(z_0, \dots, z_{p-1}), z_i \in [0, Z_i], i = 0, \dots, p-1\}$, в котором все $Z_i + 1$ являются простыми сомножителями в разложении числа $\prod_{j=0}^{s-1} (X_j + 1)$. По доказанному выше, существу-

ет биекция между R_X^s и R_Z^p , а также биекция между R_Z^p и R_Y^t , что означает наличие биекции между R_X^s и R_Y^t . Таким образом, множества R_X^s и R_Y^t всегда изоморфны при выполнении условия теоремы. $\square$

Для алгоритма изменения размерности датасета воспользуемся отображением, построенным при доказательстве Теоремы 1 и будем повышать (понижать) размерность на 1 на каждом шаге. Алгоритм выглядит следующим образом:

Вход: Датасет размерности J со значениями признака j в интервале от 0 до X_j .

Выход: Датасет размерности K со значениями признака k в интервале от 0 до Y_k , список параметров перехода.

Процедура:

1. Если $K > J$, перейти к шагу 8.
2. Среди величин X_j ($j = 0, \dots, J-1$), характеризующих датасет, выбрать минимальную. Если их несколько — выбрать ту, у которой минимальный индекс (номер признака). Результат обозначим X_{j_1} .
3. Повторить шаг 2 для набора величин X_j без X_{j_1} , получить X_{j_2} .
4. Объединить признаки j_1 и j_2 по формуле: $y_{j_1} = (X_{j_2} + 1)x_{j_1} + x_{j_2}$.
5. Положить $y_0 = x_0, \dots, y_{j_1-1} = x_{j_1-1}, y_{j_1+1} = x_{j_1+1}, \dots, y_{j_2-1} = x_{j_2-1}, y_{j_2} = x_{j_2+1}, \dots, y_{J-2} = x_{J-1}$.
6. Записать набор из четырех чисел ($j_1, X_{j_1} + 1, j_2, X_{j_2} + 1$) в список параметров перехода.
7. Если размерность полученного датасета равна K , выдать датасет и список параметров перехода. Иначе — повторить шаги 2–6 для полученного после шага 5 датасета.
8. Среди величин X_j выбрать максимальную. Если их несколько — выбрать ту, у которой максимальный индекс (номер признака). Результат обозначим X_k .

9. Если $X_k + 1$ составное число, то факторизовать его фактором, наиболее близким к $\sqrt{X_k + 1}$, получить $Y_k + 1, Y_J + 1$.

10. Если $X_k + 1 > 2$ простое число, то факторизовать $X_k + 2$ фактором, наиболее близким к $\sqrt{X_k + 2}$, получить $Y_k + 1, Y_J + 1$. Иначе выдать полученный датасет и список параметров с формулировкой “дальнейшее увеличение размерности невозможно”.

11. Разделить признак k на два признака по формулам: $y_k = x_k \bmod (Y_k + 1)$, $y_J = x_k \setminus (Y_k + 1)$.

12. Положить $y_0 = x_0, \dots, y_{k_1-1} = x_{k_1-1}$, $y_{k_1+1} = x_{k_1+1}, \dots, y_{J-1} = x_{J-1}$.

13. Записать набор из четырех чисел (k, Y_k, J, Y_J) в список параметров перехода.

14. Если размерность полученного датасета равна K , выдать датасет и список параметров перехода. Иначе — повторить шаги 8–13 для полученного датасета.

Этот алгоритм реализован в модуле `genser`: преобразование размерности “туда” осуществляется с помощью функции `transform_to`, принимающей исходный датасет и число — нужную размерность; преобразование “обратно” осуществляется функциями `transform_out_up` и `transform_out_down`, в которые передается трансформированный ранее датасет и список параметров перехода (на выходе — исходный датасет).

Теперь мы можем сравнительно вольно обращаться с размерностью датасета: ее можно трансформировать без потери информации к любой размерности от 1 до $\left\lfloor \sum_{j=0}^{s-1} \log(X_j + 1) \right\rfloor$. Например, 3-мерный датасет с $X_0 = 1$, $X_1 = 12$, $X_2 = 50$ можно преобразовать к датасету любой размерности от 1 до 10. В частности, всегда¹ можно найти такое n , для которого измененная размерность датасета будет удовлетворять условию $M = 2 \cdot 2^n - 2$ и, следовательно, данные могут быть представлены как квантовые.

2.3. Запутанный базис

Согласно постулатам квантовой механики, если существуют две системы с n_1 и n_2 кубитами, соответственно, то состояния объединенной системы имеют вид:

$$|q\rangle = \sum_{r=0}^{2^{n_1+n_2}-1} c^r |r\rangle,$$

¹ На самом деле, есть некоторые ограничения, но их можно преодолеть, меняя диапазоны значений признаков.

где c^r — амплитуды состояния системы в новом вычислительном базисе, образованном тензорным произведением исходных базисов: $|r\rangle = |k_1\rangle |k_2\rangle$, где $k_1 = 0, \dots, 2^{n_1} - 1$, $k_2 = 0, \dots, 2^{n_2} - 1$.

Получается, что если были системы с пространствами состояний $S^{2^{n_1+1}-2}$ и $S^{2^{n_2+1}-2}$, то объединенная система будет иметь пространство состояний $S^{2^{n_1+n_2+1}-2}$. Но, как легко убедиться,

$$S^{2^{n_1+1}-2} \otimes S^{2^{n_2+1}-2} \neq S^{2^{n_1+n_2+1}-2}.$$

В случае, когда $n_1 = n_2 = 1$, получим

$$S^{2^{n_1+1}-2} \otimes S^{2^{n_2+1}-2} = S^2 \otimes S^2 \neq S^6 = S^{2^{n_1+n_2+1}-2}.$$

Причем в этом простейшем случае причина очевидна — при объединении систем фазовая мультипликативная инвариантность остается только с одним параметром, т.е., добавляется еще один параметр, который раньше был “скрытым” — это был фазовый множитель в одной из исходных систем. Но эта причина не до конца объясняет несовпадение пространств состояний: она добавляет только S^1 , но снова

$$S^2 \otimes S^2 \otimes S^1 \neq S^6$$

ни по размерности, ни топологически. Иными словами, всегда найдется такое состояние объединенной системы, которое не получается произведением состояний исходных систем: налицо системное свойство объединенной системы.

Часть системы $n_1 + n_2$ -кубитов, которая не может быть выражена как произведение состояний подсистем, образует набор так называемых *запутанных состояний*. Основное свойство запутанного состояния заключается в том, что для вывода системы из запутанного состояния необходимо выполнить унитарное преобразование, которое существенно влияет на все ее подсистемы. В рассмотренном выше примере системы двух кубитов состояниями, которые не разлагаются в композиции состояний кубитов, являются *состояния Белла*:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \quad |\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

Как известно, они образуют *базис Белла*. Построение таких состояний легко обобщается на более высокие размерности — состояния в них остаются запутанными и тоже образуют базис в своем пространстве, далее он будет называться *запутанным базисом*. Его построение осуществляется схемой, обратной схеме на рис. 1.

Если состояние многокубитной системы запутано, то невозможно выйти из него, не затронув

каждый кубит. В то же время каждое состояние системы может быть записано в запутанном базисе. Это означает, что каждый компонент состояния в этом базисе существенно влияет на все кубиты. Если мы измерим амплитуды этих компонентов, то сможем увидеть, как подсистемы взаимодействуют в этой квантовой системе. Измерение — многомерный аналог *измерения Белла* — представлено на рис. 1. Если метки состояний заданы, то необходимо определить, какой базисный вектор соответствует интересующей нас метке, что может быть определено из статистики результатов измерений для данной метки. Далее, если измерять новые состояния этой системы, то с определенной вероятностью можно предсказать их принадлежность к помеченному классу, что соответствует результату измерения, который в наибольшей степени относится к помеченным образцам.

В определенной степени это означает, что классификация может быть выполнена без оптимизации, если зависимости уже есть в данных. Но эти зависимости должны существовать: например, когда данные представляют собой полную суперпозицию чистых состояний, классификация завершится неудачей, так как для каждой метки результаты измерений будут примерно равновероятны.

2.4. Квантовая нейронная сеть на запутанных состояниях

В работе [18] представлены примеры классификации на запутанных состояниях как с оптимизацией, так и без нее. Для настоящего исследования требуется кратко описать построение квантовой нейронной сети на классификаторах без оптимизации, рассматриваемых как квантовые нейроны.

Можно сказать, что на рис. 1 представлен квантовый нейрон (далее *q-нейрон*), если убрать измерение в первом регистре. В этом случае на входе будет состояние n -частичной системы, а на выходе — состояние кубита и $n - 1$ -значное измерение. Чтобы построить *квантовую нейронную сеть* (далее *КНС*), заметим, что q -нейрон, принимая на вход n кубитов, возвращает $m < n$ кубитов вместе с $k = n - m$ классическими битами информации. КНС может быть обучена на основе этой классической информации и заданной разметки данных. Процедура обучения в случае q -нейронов без оптимизации несколько отличается от обычной: не нужно никаких итераций (эпох обучения), а требуются лишь данные и их метки. Причем, если метка экземпляра данных пришла уже после проведенной классификации, то характеристики КНС, представляющие собой сопоставления базисных векторов меткам, могут измениться прямо в этот момент. То есть такое

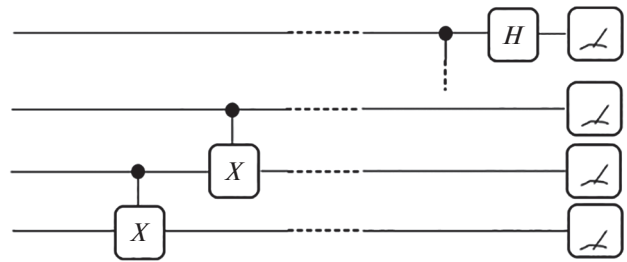


Рис. 1. Схема классификатора без оптимизации: во входящих состояниях должны быть зависимости.

обучение автоматически становится онлайн-обучением.

Копирование квантовых состояний запрещено. Поэтому КНС не содержат разветвлений. Но q -нейрон может содержать более одного квантового выхода — для этого достаточно просто убрать измерения в нужном числе регистров. Это можно использовать для создания сетей различной архитектуры. Так, КНС прямого распространения получается, если выходы нескольких q -нейронов становятся входами для следующего q -нейрона. А для создания рекуррентной КНС, нужно выход одного или нескольких q -нейронов объединить с новыми квантовыми данными для подачи в следующий слой.

Во всех КНС будет выполняться правило: сумма числа Q кубитов на выходе сети и числа C классических битов, полученных в результате измерений, равна числу I входящих кубитов:

$$I = Q + C.$$

Однако, для сети прямого распространения, все I кубитов должны быть введены в сеть сразу, а для рекуррентной сети квантовые данные могут вводиться постепенно. То есть для рекуррентной КНС можно установить какое-то предельное число C классических бит, используемых в обучении, и работать со стримом квантовых данных, не загружая расчеты избытком статистических данных.

3. РЕЗУЛЬТАТЫ

Пусть имеется некоторый набор данных $\{d_j^i, l_j\}$, где d_j^i есть значение i -го признака в j -й выборке, а l_j — значение метки (класса) для j -й выборки. Разделим все данные на обучающую и тестовую выборки и обозначим их

$$\{d_{ji}^i, l_{ji}^i\} \quad \text{и} \quad \{d_{jc}^i, l_{jc}^i\},$$

соответственно. Будем считать, что размерность данных уже удовлетворяет условию $M = 2 \cdot 2^n - 2$, метки принимают значения от 0 до $L - 1$, а значе-

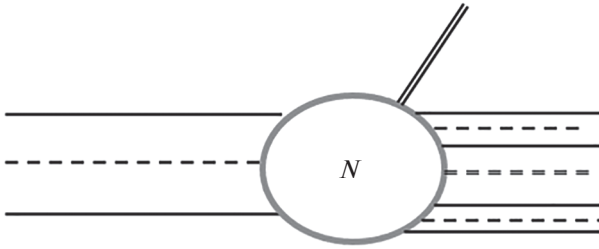


Рис. 2. Блок КНС прямого распространения.

ния всех признаков уже находятся в нужных интервалах:

$$\delta_j^i \equiv d_j^i \in \left[0, \frac{\pi}{2}\right], \quad i = 0, \dots, \frac{M}{2} - 1$$

$$\gamma_j^{\frac{i-M}{2}} \equiv d_j^i \in [0, 2\pi), \quad i = \frac{M}{2}, \dots, M - 1.$$

Будем считать, что $L = 2^l$.

Закодируем данные в квантовые состояния следующим образом:

$$z_j^0 = \cos \delta_j^0 \cos \delta_j^1 \dots \cos \delta_j^{\frac{M}{2}-1},$$

$$z_j^1 = \sin \delta_j^0 \cos \delta_j^1 \dots \cos \delta_j^{\frac{M}{2}-1} e^{i\gamma_j^0} \quad (4)$$

$$z_j^2 = \sin \delta_j^1 \cos \delta_j^2 \dots \cos \delta_j^{\frac{M}{2}-1} e^{i\gamma_j^1}, \dots, z_j^{\frac{M}{2}} = \sin \delta_j^{\frac{M}{2}-1} e^{i\gamma_j^{\frac{M}{2}-1}}$$

$$|q_j\rangle = \sum_{k=0}^{\frac{M}{2}} z_j^k |k\rangle. \quad (5)$$

Далее рассмотрим отдельно обучение КНС прямого распространения и рекуррентной КНС.

3.1. Онлайн обучение в КНС прямого распространения

Рассмотрим КНС, один блок которой представлен на рис. 2.

На рис. 2, как обычно, квантовые состояния обозначены сплошными одиночными линиями, а классические выходы — двойными линиями. Пунктиром показано повторение регистров и групп регистров. Буквой N обозначен блок, представляющий собой квантовую схему, представленную на рис. 1, в которой измерения установлены не во всех регистрах: часть верхних регистров остается неизмеренной и выходит квантовыми состояниями. В качестве структуры данных для такого блока будем использовать следующую:

$$B_k^s = \left(|q_k^s\rangle, v_k^s, \{n_k^s : |q_{n_k^s}^{s+1}\rangle \right).$$

Здесь s — номер слоя КНС, k — номер q -нейрона в слое. На входе в блок имеем квантовое состояние $|q_k^s\rangle$, после прохождения этого q -нейрона получается число v_k^s (результат измерения) и набор квантовых состояний $|q_{n_k^s}^{s+1}\rangle$, каждое из которых идет на вход n_k^s -го q -нейрона следующего слоя. Исходящие из слоя s состояния $|q_{n_k^s}^s\rangle$ объединяются в квантовое состояние

$$|q_n^{s+1}\rangle = |q_{n_0^s=n}^s\rangle \dots |q_{n_{K_s-1}^s=n}^s\rangle,$$

где K_s — число q -нейронов в слое s , так, что при выполнении условия в нижнем индексе, состояние включается в состояние $|q_n^{s+1}\rangle$, а иначе — не включается. Так образуются все блоки B_k^s для всех слоев КНС. В сети прямого распространения в последнем слое измерим все регистры. Величины v_k^s , которые получаются в результате измерений, можно объединить в одно число, двоичная запись которого будет иметь вид

$$v = v^S v^{S-1} \dots v^1,$$

где v^S — это последовательная запись измерения регистров в последнем слое (номер S), и так далее. Таким образом, в случае КНС прямого распространения, разрядность числа v будет равна числу входящих в схему одночастичных регистров. При этом для хорошего качества классификации необходимо, чтобы было $\log(\max v + 1) > l$.

Параметры обучения (аналоги весов) в данном случае будут составлять матрицу K_{va} , элементы которой равны количеству случаев получения при измерении значения v при настоящей метке a . Далее, для осуществления собственно классификации, найдем частоты меток a при полученном значении v :

$$v_{va} = \frac{K_{va}}{\sum_a K_{va}}. \quad (6)$$

и шкалируем их на единичный отрезок:

$$\kappa_{va} = \frac{v_{va} - \min_v v_{va}}{\max_v v_{va} - \min_v v_{va}}. \quad (7)$$

Для использования при классификации построим словарь D , в котором каждому ключу — предполагаемой метке — сопоставим список значений — чисел v — таких, при которых нормированная частота κ_{va} превышает частоту появления самой распространенной метки в обучающей вы-

борке (возможен вариант “превышает среднюю частоту меток в выборке”).

$$D = \{a : [v^{a_0}, \dots, v^{a_{t_a}}], \forall a = 0, \dots, L - 1\},$$

где t_a — количество значений v , выбранных по критерию для данного a . Словарь D дает возможность сопоставить метку пробному экземпляру данных после его прохождения через КНС: меткой будет тот ключ в словаре D , в списке которого окажется полученное при измерении число.

Алгоритм обучения будет следующим.

Вход: обучающие данные; нулевая матрица K_{va} ; правило отбора значений v для ключа a в словарь D .

Выход: словарь D , матрица K_{va} .

Процедура:

1. Пропустить экземпляр данных через КНС и получить результат измерений v .
2. Получить истинную метку a этого экземпляра данных.
3. Увеличить значение K_{va} на 1.
4. Повторить шаги 1–3 для всех экземпляров обучающих данных.
5. Вычислить все κ_{va} по формулам (6) и (7).
6. Для каждого значения метки a перебрать все возможные значения v и, при выполнении правила отбора, занести значение v в список по ключу a словаря D .

Алгоритм работы классификатора на тестовых данных.

Вход: экземпляр данных без метки; словарь D , матрица K_{va} .

Выход: метка экземпляра данных или *None*.

Процедура:

1. Пропустить экземпляр данных через КНС и получить результат измерений v .
2. Выбрать ключ a , по которому v находится в списке в словаре D .
3. Если шаг 2 дал результат, то выдать a , иначе выдать *None*.

Если через какое-то время истинная метка для тестового экземпляра данных становится известной, то этот экземпляр вместе с меткой подают на вход алгоритма обучения. Тем самым достигается онлайн-обучение: обучающие данные корректируются по мере прихода новых данных с разметкой.

Рассмотрим эти алгоритмы на примере датасета о задержках авиарейсов, взятого с ресурса [kaggle.com](https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay)². Исходный датасет содержит 7 признаков и одну метку. Признаки: авиакомпания, рейс,

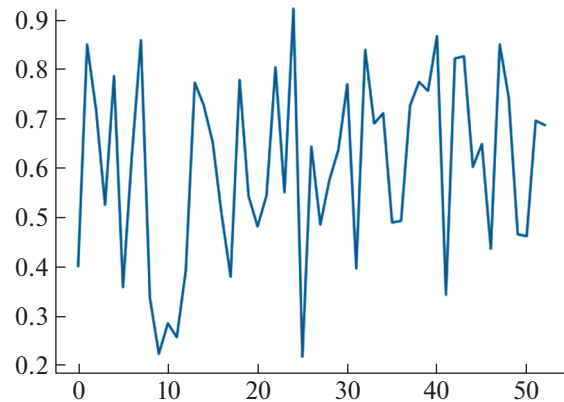


Рис. 3. Значения метрики F1 в 53 последовательных тестах.

аэропорт отправления, аэропорт назначения, день недели, время. Метка — наличие задержки. В датасете 539383 экземпляра данных, из которых 240264 имеют метку 1, остальные — 0. Все значения признаков были переведены в целые числа и размерность датасета была обратимым образом увеличена до 30 — это позволило представлять данные в виде состояния 4-частичной квантовой системы. Для таких входящих состояний можно сделать либо однослойную КНС с единственным q -нейроном, либо двухслойную с двумя q -нейронами в первом слое и одним во втором. Рассмотрим двухслойную КНС.

Очевидно, $v = 0, \dots, 15$, так как на входе 4-частичное состояние. В этом датасете экземпляры данных упорядочены по времени. Поэтому поставим задачу так: будем прогнозировать задержки 1000 рейсов на основе задержек предыдущих 1000 рейсов. Таким образом, обучающая выборка у нас будет включать 1000 экземпляров данных. Реализуем обучающий и тестовый алгоритмы и получим метрику F1 для каждого теста. Очевидно, таких тестов в этом датасете возможно сделать 538, но мы ограничимся 53 — этого будет достаточно, чтобы увидеть тенденцию. Как видно на графике, представленном на рис. 3, значение метрики сильно изменяется от теста к тесту. Однако, во-первых, среднее значение F1 почти в два раза превышает частоту задержек в полной обучающей выборке (из 53 000 экземпляров) и, во-вторых, снижение качества классификации наблюдается с примерной периодичностью, что может говорить о накоплении неучтенных долгосрочных факторов. При использовании рекуррентной КНС это обстоятельство должно быть хотя бы частично преодолено.

² <https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay> (просмотрено 13.08.2023)

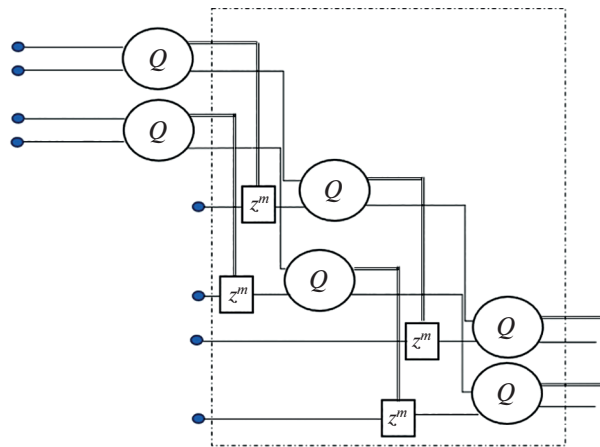


Рис. 4. Рекуррентная КНС для 4-частичного входа.

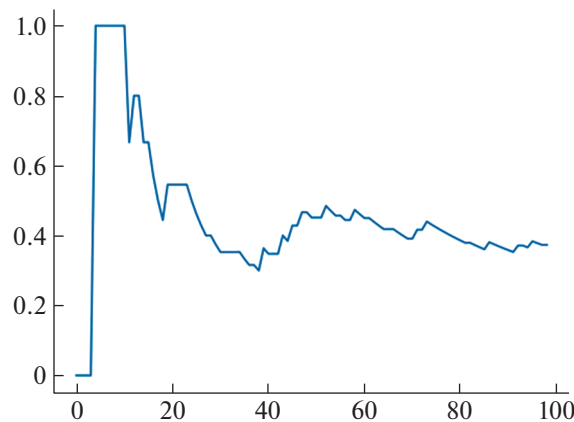


Рис. 5. Значения метрики F1 при последовательных проходах данных с классификацией последнего экземпляра (размер пакета 5000 экземпляров).

3.2. Онлайн обучение в рекуррентной КНС

Обратимся теперь к рекуррентной КНС, онлайн обучаемой без оптимизации. В отличие от предыдущего случая сети прямого распространения, теперь можно запускать экземпляры данных, относящиеся к разным моментам времени, в разные слои сети. Алгоритм обучения при этом будет основан на тех же соображениях, что и раньше, но на вход его будут поданы экземпляры размеченных данных $(|q_t\rangle, a)$ с имеющимися уникальной временной меткой t и классом a . Схема рекуррентной КНС для рассматриваемого датасета авиарейсов представлена на рис. 4. Количество входящих экземпляров данных может быть почти произвольным — ограничено только числом имеющихся кубитов и количеством имеющихся экземпляров данных. Справа от схемы рис. 4 будет стоять такое же завершение, как в случае КНС прямого распространения. И на выходе этой сети будет 4-разрядное двоичное число.

Для эксперимента было выбрано число последовательных векторов, входящих в рекуррентную сеть, равное 5000. Может показаться, что для этого потребуются тысячи кубитов, но это не так: достаточно тех 12 кубитов, что использованы на схеме рис. 4, поскольку обведенная контуром часть схемы может повторяться на тех же кубитах, устанавливаемых каждый раз в требуемое состояние.

Рекуррентная сеть в эксперименте использовалась для предсказания задержки последнего рейса из пакета векторов, т.е., каждого 5000-го рейса. Сеть обучалась накопительным итогом (накапливались значения в матрице K).

Эксперимент показал, что рекуррентная КНС показывает более качественную классификацию, чем КНС прямого распространения. В то же время, после определенного числа шагов с накоплением значений, качество начинает ухудшаться. Это явление, аналогичное переобучению.

Результат предсказания задержек рейсов с помощью рекуррентной КНС представлен на рис. 5. На нем видно, что для принятых параметров эксперимента, максимальное значение метрики F1 было достигнуто на примерно 10 проходе и дальше качество начало падать. Таким образом, имеет смысл ограничить накопление значений в матрице K 10 шагами.

Десять шагов с пакетами длиной 5000 — это обучение на истории 50 000 экземпляров. При этом на ближайшие 5–10 шагов сеть дает качество предсказания, близкое к 80%.

4. ЗАКЛЮЧЕНИЕ

Рассмотренное в работе квантовое машинное обучение имеет класс QC или QQ, т.е., использует квантовые данные на классических (эмулирующих) или квантовых устройствах. Предложенный q -нейрон может работать без обучения в обычном смысле: не нужны оптимизация и обратное распространение ошибки. Его обучение сводится к получению статистических весов исходов.

Построенные из q -нейронов квантовые нейронные сети предложены в двух топологиях: прямого распространения и рекуррентная. Рассмотрены процессы обучения КНС. Отдельно рассмотрен вопрос подготовки данных для анализа с помощью КНС. В частности, сформулирована и доказана Теорема 1, которая показывает возможность трансформации размерности данных в широком диапазоне значений, без потери информации (обратно). Такая трансформация является необходимой для приготовления квантовых состояний из имеющихся классических векторов данных.

Новое направление развития квантового машинного обучения, предложенное в статье [18], использующее квантовую запутанность, в настоящей работе развито и получило дальнейшее теоретическое обоснование и практическое подтверждение.

Системы искусственного интеллекта, основанные на предложенных в настоящей работе конфигурациях квантовых нейронных сетей, далее планируется применять для создания интеллектуальных адаптивных систем управления автономными мобильными роботами, функционирующими в различных средах. Ключевой особенностью разрабатываемых систем является способность к быстрому онлайн-обучению, не требующему функциональной оценки (онлайн-обучение без подкрепления). В случае малых размерностей данных такие системы могут обойтись без квантовых вычислителей.

БЛАГОДАРНОСТИ

Работа выполнена в рамках программы развития Приоритет-2030 на материальной базе Центра высоких технологий БГТУ им. В.Г. Шухова.

СПИСОК ЛИТЕРАТУРЫ

1. Tacchino F., Macchiavello C., Gerace D. et al. An artificial neuron implemented on an actual quantum processor. *npj Quantum Inf.* 2019. V. 5. № 1. P. 26.
2. Menneer T., Narayanan A. Quantum-inspired neural networks. In *Proceedings of the Neural Information Processing Systems 95*, Denver, CO, USA, 27–30 November 1995.
3. Гушанский С.М., Буглов В.Е. Квантовое глубокое обучение сверточной нейронной сети с использованием вариационной квантовой схемы. *Известия ЮФУ. Технические науки*. 2021. Т. 7. № 224. С. 167–174.
4. Cong I., Choi S., Lukin M.D. Quantum convolutional neural networks. *Nat. Phys.* 2019. V. 15. P. 1273–1278.
5. Kerenidis I., Landman J., Prakash A. Quantum algorithms for deep convolutional neural networks. *arXiv* 2019. V. arXiv. P. 1911.01117.
6. Henderson M., Shukla S., Pradhan S., Cook T. Quantum convolutional neural networks: Powering image recognition with quantum circuits. *Quantum Mach. Intell.* 2020. V. 2. P. 2.
7. Rebentrost P., Mohseni M., Lloyd S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* 2014. V. 113. P. 130503.
8. Harrow A.W., Hassidim A., Lloyd S. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.* 2009. V. 103. P. 150502.
9. Dang Y., Jiang N., Hu H., Ji Z., Zhang W. Image classification based on quantum K-Nearest-Neighbor algorithm. *Quantum Inf. Process.* 2018. V. 17. P. 1–18.
10. Schuld M., Sinayskiy I., Petruccione F. Prediction by linear regression on a quantum computer. *Phys. Rev. A*. 2016. V. 94. P. 022342.
11. Lu S., Braunstein S.L. Quantum decision tree classifier. *Quantum Inf. Process.* 2014. V. 13. P. 757–770.
12. Lloyd S., Mohseni M., Rebentrost P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* 2013. V. arXiv. P. 1307.0411.
13. Lloyd S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 1982. V. 28. P. 129–137.
14. Kerenidis I., Landman J., Luongo A., Prakash A. q-means: A quantum algorithm for unsupervised machine learning. *arXiv* 2018. V. arXiv. P. 1812.03584.
15. Aïmeur E., Brassard G., Gambs S. Quantum speed-up for unsupervised learning. *Mach. Learn.* 2013. V. 90. P. 261–287.
16. DiVincenzo D.P. The Physical Implementation of Quantum Computation. *Fortschritte der Physik*. 2000. V. 48. № 9–11. P. 771–783.
17. Lloyd S., Mohseni M., Rebentrost P. Quantum principal component analysis. *Nat. Phys.* 2014. V. 10. P. 631–633.
18. Зуев С.В. Геометрические свойства квантовой запутанности и машинное обучение. *Russian Technological Journal*. 2023. Т. 12. № 5: в печати.
19. Bruza P.D., Cole R.J. Quantum Logic of Semantic Space: An Exploratory Investigation of Context Effects in Practical Reasoning, 2006. Available online: <https://arxiv.org/abs/quant-ph/0612178> (просмотрено 20.08.2023).

STATISTICAL ONLINE LEARNING IN RECURRENT AND FEEDFORWARD QUANTUM NEURAL NETWORKS

S. V. Zuev^a

^a*Belgorod State Technological University named after V.G. Shukhov, Belgorod, Russian Federation*

Presented by Academician of the RAS A.L. Semenov

Purpose. For adaptive artificial intelligence systems, the question of the possibility of online learning is especially important, since such training provides adaptation. The purpose of the work is to consider methods of quantum machine online learning for the most common two architectures of quantum neural networks: feedforward and recurrent.

Methods. The work uses the quantumz module available on PyPI to emulate quantum computing and create artificial quantum neural networks. In addition, the genser module is used to transform data dimensions, which provides reversible transformation of dimensions without loss of information. The data for the experiments are taken from open sources. The paper implements the machine learning method without optimization, proposed by the author earlier.

Results. Online learning algorithms for recurrent and feedforward quantum neural network are presented and experimentally confirmed.

Conclusions. The proposed learning algorithms can be used as data science tools, as well as a part of adaptive intelligent control systems. The developed software can fully unleash its potential only on quantum computers, but, in the case of a small number of quantum registers, it can also be used in systems that emulate quantum computing, or in photonic computers.

Keywords: online learning, adaptive artificial intelligence, quantum machine learning, quantum entanglement