

Кубит

Семен Синченко

6 марта 2021 г.

Содержание

1	Описание лекции	2
2	Что такое кубит	2
3	Состояние vs значение	2
3.1	Состояние классического бита	2
3.2	Кот Шрёдингера	3
3.3	Состояние кубита	3
3.4	Связь состояния и значения кубита	4
3.4.1	Базисные состояния	4
3.4.2	Амплитуды вероятностей	4
4	Сфера Блоха	5
4.1	Возможные базисы	5
4.1.1	Z-базис	5
4.1.2	X-базис	6
4.1.3	Y-базис	6
4.2	Сфера Блоха	6
5	А что можно делать с таким кубитом?	7
5.1	Линейные операторы	7
5.2	Обратимость	7
5.3	Унитарность	7
5.4	Пример оператора	8
5.4.1	Гейт Адамара	8

6	Измерения	9
6.1	Операторы Паули	9
6.2	Собственные значения	10
6.2.1	Собственные вектора $\hat{\sigma}^Z$	10
6.3	Формальная запись	11
6.4	Другие операторы Паули	11
6.5	Ожидаемое значение при измерении	11
7	Что мы узнали?	12

1. Описание лекции

Из этой лекции мы узнаем:

- Что такое кубит
- В чем разница между значением и состоянием
- Что такое сфера Блоха
- Какие можно делать операции над кубитами
- Что такое измерение

2. Что такое кубит

Как мы знаем, классический компьютер оперирует двоичными числами (0 и 1). Минимальный объем информации для классического компьютера называется бит. Квантовый компьютер оперирует квантовыми битами, или кубитами. Если мы будем считывать кубит, то он также как и классический бит, будет иметь лишь два возможных значения (0 и 1). Так в чем же разница с обычным битом? В чем особенности квантовых компьютеров, которые и дают им преимущества над классическими компьютерами?

Вся суть на самом деле в том, что для квантомеханических систем (и кубитов в частности) их **состояния** и **значения** это не одно и то же!

3. Состояние vs значение

3.1. Состояние классического бита

Обычно мы не отличаем состояние классического бита от его значения и считаем, что если бит имеет значение 1, то и состояние его описывается числом

1.

3.2. Кот Шрёдингера

Давайте вспомним известный мысленный эксперимента Шрёдингера – кот, который одновременно и жив и мертв. Понятно, что **значение** кота точно одно - он либо жив, либо мертв. Но **состояние** его более сложное – он находится в **суперпозиции** состояний жив и мертв одновременно!

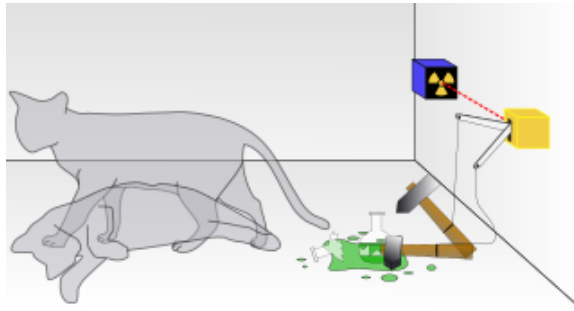


Рис. 1: Кот Шрёдингера

3.3. Состояние кубита

Состояние кубита, если можно так сказать, аналогично коту Шрёдингера: оно отличается от значения кубита и описывается вектором из двух комплексных чисел. Мы будем обозначать состояния (или вектора) символом $|\Psi\rangle$ (кет – вектор-столбец) – это широко принятая в квантовой механике и квантовых вычислениях нотация Дирака:

$$|\Psi\rangle = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Что именно значат числа c_1 и c_2 мы поговорим чуть далее, а пока запишем наш кубит $|\Psi\rangle$ в коде Python. Возьмем, для начала, $c_1 = c_2 = \frac{1}{\sqrt{2}}$.

```
import numpy as np
```

```
qubit = np.array([1 / np.sqrt(2) + 0j, 1 / np.sqrt(2) + 0j]).reshape((2, 1))
```

Заметьте, что здесь мы создаем именно вектор-строку размерности 2×1 .

```
qubit.shape  
# (2, 1)
```

3.4. Связь состояния и значения кубита

А теперь разберем подробнее, что же такое наш вектор $|\Psi\rangle$ и что же значат цифры c_1, c_2 . Для начала посмотрим на то, а какие состояния имеет кубит, для которого мы точно знаем значение. То есть посмотрим на кота Шрёдингера, но который точно жив или точно мертв.

3.4.1. Базисные состояния

Посмотрим, как выглядят состояния кубитов со точно определенными значениями:

$$|0\rangle = \begin{bmatrix} 1 + 0j \\ 0 + 0j \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 + 0j \\ 1 + 0j \end{bmatrix}$$

Что мы можем сказать об этих состояниях? Как минимум следующее:

- Они ортогональны ($|0\rangle \perp |1\rangle$)
- Они имеют единичную норму
- Они образуют базис

Что это значит для нас? А то, что любое состояние $|\Psi\rangle$ мы можем записать как линейную комбинацию векторов $|0\rangle$ и $|1\rangle$, причем коэффициенты в этой комбинации будут как раз наши c_1, c_2 :

```
basis_0 = np.array([1 + 0j, 0 + 0j]).reshape((2, 1))  
basis_1 = np.array([0 + 0j, 1 + 0j]).reshape((2, 1))  
  
c1 = c2 = 1 / np.sqrt(2)  
  
np.allclose(qubit, c1 * basis_0 + c2 * basis_1)  
# True
```

3.4.2. Амплитуды вероятностей

Квантовая механика устроена таким интересным образом, что если мы будем измерять **значение** кубита, то вероятность каждого из вариантов будет

пропорциональная соответствующему коэффициенту в разложении **состояния**. Но так как амплитуды у нас это в общем случае комплексные числа, а вероятности должны быть строго действительные, мы должны домножить амплитуды на комплексно сопряженные значения. В случае наших значений $c_1 = c_2 = \frac{1}{\sqrt{2}}$ мы получаем:

```
p0 = np.conj(c1) * c1
p1 = np.conj(c2) * c2

np.allclose(p0, p1)
# True

np.allclose(p0 + p1, 1.0)
# True
```

Тут мы сразу видим еще одну важную вещь – сумма вероятностей всех состояний должна быть равна 100%. Это сразу приводит нас к тому, что состояния это не любые комплексные вектора, а комплексные вектора с единичной нормой:

```
np.allclose(np.conj(qubit).T @ qubit, 1.0)
# True
```

Мы будем очень часто пользоваться транспонирование и взятием комплексно сопряженного от векторов и в квантовой механике это имеет специальное обозначение $\langle \Psi | = \Psi^{T\dagger}$ (бра – вектор-строка). Тогда наше правило нормировки из NumPy кода может быть записано в нотации Дирака так:

$$\langle \Psi | \Psi \rangle = 1$$

4. Сфера Блоха

Надо понимать, что описанный нами выше базис $|0\rangle, |1\rangle$ не является единственно возможным базисом. Наши вектора $|0\rangle, |1\rangle$ это лишь самый часто применимый базис, который называют **Z** базисом. Но есть и другие варианты.

4.1. Возможные базисы

4.1.1. Z-базис

Уже описанные нами $|0\rangle$ и $|1\rangle$.

4.1.2. X-базис

Базисные состояния $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ и $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$:

```
plus = (basis_0 + basis_1) / np.sqrt(2)
minus = (basis_0 - basis_1) / np.sqrt(2)
```

4.1.3. Y-базис

Базисные состояния $|R\rangle = \frac{|0\rangle+i|1\rangle}{\sqrt{2}}$ и $|L\rangle = \frac{|0\rangle-i|1\rangle}{\sqrt{2}}$:

```
R = (basis_0 + 1j * basis_1) / np.sqrt(2)
L = (basis_0 - 1j * basis_1) / np.sqrt(2)
```

Легко можно убедиться, что все для каждого из этих базисов вектора ортогональны:

```
np.allclose(np.conj(basis_0).T @ basis_1, 0)
# True
np.allclose(np.conj(plus).T @ minus, 0)
# True
np.allclose(np.conj(R).T @ L, 0)
# True
```

Заметьте, что в наших векторных пространствах скалярное произведение это $ab = \langle a|b\rangle$! Именно поэтому мы должны делать транспонирование и комплексное сопряжение первого вектора в паре.

4.2. Сфера Блоха

Обозначения $|0\rangle, |1\rangle, |+\rangle, |-\rangle, |R\rangle, |L\rangle$ выбраны не случайно – они имеют геометрический смысл.

Принято считать, что ось **Z** это основная ось, так как физически, квантовые компьютеры измеряют именно по этой оси. Ось **X** она, если можно так выразиться, смотрит на нас и поэтому обозначается $|+\rangle$ и $|-\rangle$. А ось **Y** направлена как бы вдоль и поэтому базис обозначают как «право» ($|R\rangle$) и «лево» ($|L\rangle$).

Вектор состояния кубита также называют волновой функцией и этот вектор может идти в любую точку сферы Блоха. Так как сама сфера имеет единичный радиус, это гарантирует нам, что для всех состояний сумму квадратов амплитуд будет равна единице.

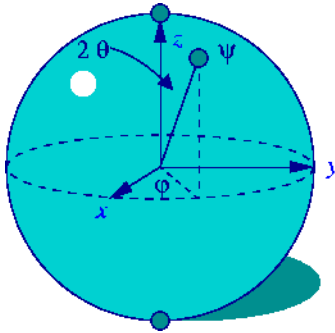


Рис. 2: Сфера Блоха

5. А что можно делать с таким кубитом?

5.1. Линейные операторы

Любое наше действие, которое мы совершаем с кубитом в состоянии $|\Psi\rangle$ должно переводить его в другое состояние $|\Phi\rangle$. Что переводит один вектор в другой вектор в том же пространстве? Правильно, матрица! Ну или, другими словами, линейный оператор. Мы будем обозначать их \hat{U} .

5.2. Обратимость

Наша природа устроена таким образом, что почти все действия, которые мы совершаем с кубитом, должны быть обратимыми. То есть для каждого действия и операции (кроме измерений, но об этом чуть дальше), например \hat{U} , должно быть такое противодействие \hat{U}^\dagger , что оно вернет кубит в первоначальное состояние:

$$\hat{U}^\dagger = \hat{U}^T$$

Тут \dagger обозначает взятие комплексно сопряженного. А такие операторы называются самосопряженными.

5.3. Унитарность

Как мы говорили, норма волновой функции должна быть нормирована на единицу, а значит любой оператор, который переводит одно состояние в другое $\hat{U}|\Psi\rangle = |\Phi\rangle$ должен сохранять нормировку, то есть должен быть **унитарным**. Более того, это свойство приводит к тому, что матрица сохраняет скалярное произведение:

$$\langle \Psi | \hat{U}^\dagger \hat{U} | \Psi \rangle = \langle \Psi | \Psi \rangle$$

Другими словами, унитарный оператор удовлетворяет условию $\hat{U}^\dagger \hat{U} = \hat{I}$

5.4. Пример оператора

Большое число операторов мы будем разбирать в дальнейших лекциях, так как именно операторы (или квантовые **гейты**) являются основой квантовых вычислений. А пока мы посмотрим один простой пример – оператор Адамара (Hadamard gate) – оператор, который переводит $|0\rangle \rightarrow |+\rangle$.

5.4.1. Гейт Адамара

Начнем с того, что пока у нас лишь один кубит. Состояние одного кубита это вектор размерности два, а значит оператор, который переводит вектор размерности два в другой вектор размерности два это матрица 2×2 . Запишем оператор Адамара в матричном виде, а потом убедимся, что он эрмитов, унитарный, а еще, что он действительно переводит состояние $|0\rangle \rightarrow |+\rangle$.

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

1. Реализация в Python

```
h = 1 / np.sqrt(2) * np.array([
    [1 + 0j, 1 + 0j],
    [1 + 0j, 0j - 1j]
])
```

2. Самосопряженность

```
np.allclose(np.conj(h), h.T)
# True
```

3. Унитарность

```
np.allclose(np.conj(h) @ h, np.eye(2))
# True
```

4. Правильное действие

```
np.allclose(h @ basis_0, plus)
# True
```


6. Измерения

Измерения в квантовых вычислениях стоят отдельно именно потому, что они «открывают» коробку с котом Шрёдингера – мы точно узнаем, жив он, или мертв. Вся **суперпозиция** его состояния исчезает.

Это интересный факт – исчезновение суперпозиции многим кажется парадоксом, именно поэтому и появляются разные интерпретации квантовой механики, например, многомировая интерпретация Эверетта. Действительно, это кажется немного странным, что полностью обратимая квантовая механика и непрерывная динамика волновых функций вдруг "ломаются" и мы получаем такой коллапс, который еще называют редукцией фон Неймана. Доктору Эверетт тоже это не нравилось и он предложил другую интерпретацию этого процесса. Согласно его теории, когда мы производим измерения мы как бы "расщепляем" нашу вселенную на две ниточки – в одной кот остается жив, а в другой остается мертв.

К сожалению, любые такие теории все равно остаются на уровне спекуляций, так как почти невозможно придумать эксперимент, который бы подтверждал или опровергал такую гипотезу. Это скорее вопрос того, как каждый из нас понимает этот процесс, так как математически все такие теории в итоге дают один и тот же результат, который мы можем наблюдать и измерять.

Как мы уже говорили, у кубита может быть несколько разных базисов: $|0\rangle, |1\rangle, |+\rangle, |-\rangle, |R\rangle, |L\rangle$. **Значение** кубита в каждом из этих базисов может быть измерено. Но что такое измерение? Как это записать математически?

6.1. Операторы Паули

На самом деле, любая наблюдаемая величина соответствует какому-то оператору. Например, измерения в разных базисах X, Y, Z соответствуют операторам Паули:

$$\hat{\sigma}^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \hat{\sigma}^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \hat{\sigma}^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

```
pauli_x = np.array([[0 + 0j, 1 + 0j], [1 + 0j, 0 + 0j]])
pauli_y = np.array([[0 + 0j, 0 - 1j], [0 + 1j, 0 + 0j]])
pauli_z = np.array([[1 + 0j, 0 + 0j], [0 + 0j, 0j - 1j]])
```

Эти операторы являются очень важными – рекомендуется знать их наизусть, так как они встречаются в каждой 2-й статье по квантовым вычислениям.

ям, а также постоянно фигурируют в документации всех основных библиотек для квантового машинного обучения!

6.2. Собственные значения

Хорошо, мы поняли, что есть какая-то связь между нашими измерениями и операторами. Но какая именно? Что значит, например, что измерения по оси Z соответствуют оператору $\hat{\sigma}^Z$?

И тут мы приходим к собственным значениям операторов. Оказывается (так устроен наш мир), что **измеряя** какую-то величину в квантовой механике мы всегда будем получать одно из собственных значений соответствующего оператора, а состояние будет коллапсировать в соответствующий собственный вектор этого оператора. Другими словами, **измеряя** кота Шрёдингера мы будем получать значения жив или мёртв, а состояние кота будет переходить в состояние, соответствующее одному из этих значений. А еще **измерение** не является обратимой операцией – однажды открыв коробку с котом и увидев, что он жив или мертв, мы уже не сможем закрыть ее обратно и вернуть кота в суперпозицию.

Важно понимать, что это не просто что-то из квантовой физики – понимание этих вещей нам очень пригодится потом, когда мы будем говорить о решении практических комбинаторных задач, таких, например, как задача о выделении сообществ в графе!

6.2.1. Собственные вектора $\hat{\sigma}^Z$

Вернемся к нашему оператору $\hat{\sigma}^Z$. Легко убедиться, что его собственные значения равны 1 и -1, а собственные соответствующие им собственные вектора это $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ и $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$:

```
np.linalg.eig(pauli_z)
# (array([ 1.+0.j, -1.+0.j]),
# array([[1.+0.j, 0.+0.j],
#        [0.+0.j, 1.+0.j]]))
```

Таким образом, измерение по оси Z всегда будет давать нам одно из этих двух значений и переводить состояние кубита в соответствующий собственный вектор.

6.3. Формальная запись

Формально мы можем записать это для любого эрмитова оператора \hat{U} , что собственные состояния этого оператора являются его собственными векторами, а собственные значения в этом случае являются наблюдаемыми значениями:

$$\hat{U}|\Psi\rangle = u|\Psi\rangle$$

6.4. Другие операторы Паули

Убедимся, что у остальных операторов Паули собственные значения такие же:

```
np.linalg.eig(pauli_x)
# (array([ 1.+0.j, -1.+0.j]),
# array([[ 0.70710678-0.j,  0.70710678+0.j],
#        [ 0.70710678+0.j, -0.70710678+0.j]]))

np.linalg.eig(pauli_y)
# (array([ 1.+0.j, -1.+0.j]),
# array([[ -0.70710678j,  0.70710678+0.j],
#        [ 0.70710678+0.j,  0.70710678j]]))
```

Можно заметить, что у всех операторов Паули нет ни одного общего собственного вектора. Таким образом, мы приходим к ситуации, когда мы не можем одновременно точно измерить два разных оператора, так как наше измерение должно переводить состояние в соответствующий собственный вектор. В квантовой механике это называется принципом неопределенности.

6.5. Ожидаемое значение при измерении

Мы не будем писать полный симулятор кубитов, который включает измерения – это требует введения сложного случайного процесса. Но мы можем легко ответить на другой вопрос. А именно, мы можем сказать, а какое будет **ожидаемое** значение оператора \hat{U} для состояния Ψ ? Ну или, другими словами, какое будет математическое ожидание большого числа измерений. Это можно записать следующим образом:

$$\mathbf{E}(\hat{U}) = \langle \Psi | \hat{U} | \Psi \rangle$$

Например, оператор $\hat{\sigma}^z$ полностью не определен в состоянии $|+\rangle$, то есть мы будем равновероятно получать значения -1 и 1, а математическое ожидание будет равно нулю:

```
plus.conj().T @ pauli_z @ plus  
# array([[0.+0.j]])
```

С другой стороны, измеряя состояние $|+\rangle$ в X -базисе мы всегда будем получать 1:

```
plus.conj().T @ pauli_x @ plus  
# array([[1.+0.j]])
```

7. Что мы узнали?

- Состояние и значение для кубита не одно и то же
- Состояния – вектора в Гильбертовом пространстве
- Квантовые операторы унитарные и самосопряженные
- Измеряемые значения – собственные значения операторов
- Измерение ломает суперпозицию