Гейты

Семен Синченко

6 марта 2021 г.

Содержание

1. Описание лекции

Из этой лекции мы узнаем:

- Какие есть основные однокубитные гейты
- Как записывать многокубитные состояния
- Какие есть многокубитные гейты
- Конструирование многокубитных операторов

2. Основные однокубитные гейты

В прошлый раз мы познакомились с операторами Паули, а также гейтом Адамара. Теперь давайте посмотрим какие еще однокубитные гейты часто применяются в квантовых вычислениях и квантовом машинном обучении.

2.1. Гейты поворота вокруг оси

Повротные гейты играют центральную роль в квантовом машинном обучении. Вспомним на секунду, как выглядят наши однокубитные состояния на сфере Блоха:

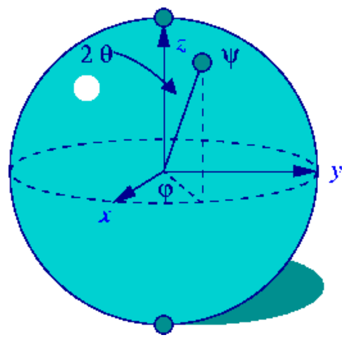

Рис. 1: Сфера Блоха

Гейты $R\hat{X}(\phi), R\hat{Y}(\phi), R\hat{Z}(\phi)$ осуществляют поворот на определенный угол ϕ вокруг соответствующей оси на сфере Блоха.

Давайте внимательно рассмотрим это на примере гейта $\hat{RY}$.

2.1.1. Гейт $\hat{RY}$

Сам гейт определяется следующим образом:

$$R\hat{Y}(\phi) = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) & -\sin\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) & \cos\left(\frac{\phi}{2}\right) \end{bmatrix}$$

```
import numpy as np

def ry(state, phi):
    return np.array([
        [np.cos(phi / 2), -np.sin(phi / 2)],
        [np.sin(phi / 2), np.cos(phi /2)]
    ]) @ state
```

Запишем наше состояние $|0\rangle$

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

Внимательно посмотрев на сферу Блоха, можно заметить, что повернув состояние из $|0\rangle$ на π и измерив значение $\hat{\sigma^z}$ мы получим 1, а повернув на $-\pi$ 0:

```
def expval(state, op):
    return state.conj().T @ op @ state

pauli_x = np.array([[0 + 0j, 1 + 0j], [1 + 0j, 0 + 0j]])

np.allclose(expval(ry(basis, np.pi / 2), pauli_x), 1.0)
# True

np.allclose(expval(ry(basis, -np.pi / 2), pauli_x), -1.0)
# True
```

Убедимся также, что вращение на угол, пропорциональный 2π не меняет результата измерения. Возьмем случайное состояние:

$$|\Psi\rangle = \begin{bmatrix} 0.42 \\ \sqrt{1 - 0.42^2} \end{bmatrix}$$

```
random_state = np.array([0.42 + 0j, np.sqrt(1 - 0.42**2) + 0j]).reshape((2, 1))
```

Измерим его по осям **X** и **Z**, затем повернем его на угол 2π и измерм снова:

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

expval(random_state, pauli_z)
# array([[-0.6472+0.j]])
expval(random_state, pauli_x)
# array([[0.76232025+0.j]])

expval(ry(random_state, 2 * np.pi), pauli_z)
# array([[-0.6472+0.j]])
expval(ry(random_state, 2 * np.pi), pauli_x)
# array([[0.76232025+0.j]])
```

2.1.2. Другие гейты вращений

Аналогичным образом определяются гейты $\hat{RX}$ и $\hat{RZ}$:

$$R\hat{X}(\phi) = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) & -i\sin\left(\frac{\phi}{2}\right) \\ -i\sin\left(\frac{\phi}{2}\right) & \cos\left(\frac{\phi}{2}\right) \end{bmatrix} \qquad R\hat{Z}(\phi) = \begin{bmatrix} e^{-\frac{i\phi}{2}} & 0 \\ 0 & e^{\frac{i\phi}{2}} \end{bmatrix}$$

В общем случае эти гейты могут быть также записаны следующим образом:

$$\hat{R^{\sigma}} = e^{-\frac{i\phi\sigma}{2}},$$

где σ – это один из операторов Пуали $\sigma^x, \sigma^y, \sigma^Z$.

2.2. Phase-shift гейт

Другой важный гейт – это так называемый phase-shift гейт, или $\hat{U}_1$ гейт. Его матричная форма имеет следующий вид:

$$U_1\hat{(}\phi) = \begin{bmatrix} 1 & 0 \\ 0 & y^{i\phi} \end{bmatrix}$$