응용물리연구실심화실습3 (PHY3075)

2024년 10월 4주차

응용물리학과 2022006971 이민성

번역

- Qiskit 예제

  이전 수업에서는 Qiskit의 Statevector와 Operator 클래스에 대해 배우고 이를 사용하여 양자 시스템을 시뮬레이션하는 방법을 살펴보았습니다. 이번 섹션에서는 이 클래스를 사용하여 여러 시스템의 동작을 탐구할 것입니다. 먼저 이 클래스들을 임포트하고, NumPy의 제곱근 함수도 함께 사용합니다.

  ```
  from qiskit.quantum_info import Statevector, Operator
  from numpy import sqrt
  ```

  출력 없음

  텐서곱

  Statevector 클래스에는 다른 Statevector와의 텐서 곱을 반환하는 `tensor` 메서드가 있습니다.

  예를 들어, 아래 코드에서는 두 개의 상태 벡터 $|0\rangle$와 $|1\rangle$을 만들고, `tensor` 메서드를 사용하여 새로운 벡터 $|0\rangle \otimes |1\rangle$을 생성합니다.

  ```
  zero, one = Statevector.from_label("0"), Statevector.from_label("1")
  zero.tensor(one).draw("latex")
  ```

  출력:

  $|01\rangle$

  다음 예제에서는 상태 벡터 $|+\rangle$와 1/sqrt(2)($|0\rangle$+i$|1\rangle$) 상태를 나타내는 벡터를 생성하고, 이를 결합하여 새로운 상태 벡터를 만듭니다. 이 새로운 벡터를 `psi` 변수에 할당합니다.

  ```
  plus = Statevector.from_label("+")
  i_state = Statevector([1 / sqrt(2), 1j / sqrt(2)])
  psi = plus.tensor(i_state)

  psi.draw("latex")
  ```

  출력:

  $$\frac{1}{2}|00\rangle + \frac{i}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{i}{2}|11\rangle$$

  Operator 클래스의 텐서 곱

  Operator 클래스에도 `tensor` 메서드가 있습니다. 아래 예제에서는 X 게이트와 I 게이트를 생성하고, 이들의 텐서 곱을 표시합니다.

```
X = Operator([[0, 1], [1, 0]])
I = Operator([[1, 0], [0, 1]])

X.tensor(I)
```

출력:

```
Operator([[0.+0.j, 0.+0.j, 1.+0.j, 0.+0.j],
          [0.+0.j, 0.+0.j, 0.+0.j, 1.+0.j],
          [1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j],
          [0.+0.j, 1.+0.j, 0.+0.j, 0.+0.j]],
         input_dims=(2, 2), output_dims=(2, 2))
```

우리는 이제 이러한 복합 상태와 연산을 이전 수업에서 다룬 단일 시스템처럼 다룰 수 있습니다. 예를 들어, 아래 셀에서는 우리가 정의한 상태 ψ에 대해 (I⊗X) |ψ⟩를 계산합니다. (여기서 ^ 연산자는 행렬을 텐서 곱으로 결합합니다.)

```
psi.evolve(I ^ X).draw("latex")
```

출력:

$$\frac{i}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{i}{2}|10\rangle + \frac{1}{2}|11\rangle$$

다음으로, CX 연산자를 생성하고 CX |ψ⟩를 계산합니다.

```
CX = Operator(
    [
        [1, 0, 0, 0],
        [0, 1, 0, 0],
        [0, 0, 0, 1],
        [0, 0, 1, 0],
    ]
)

psi.evolve(CX).draw("latex")
```

출력:

$$\frac{1}{2}|00\rangle + \frac{i}{2}|01\rangle + \frac{i}{2}|10\rangle + \frac{1}{2}|11\rangle$$

부분 측정

이전 페이지에서는 `measure` 메서드를 사용하여 양자 상태 벡터의 측정을 시뮬레이션했습니다. 이 메서드는 두 가지 항목을 반환합니다: 시뮬레이션된 측정 결과와 이 측정에 따른 새로운 `Statevector`입니다.

기본적으로 `measure`는 상태 벡터의 모든 큐비트를 측정하지만, 정수 리스트를 제공하여 특정 인덱스의 큐비트만 측정할 수 있습니다. 이를 시연하기 위해, 아래 셀은 상태

$$W = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle).$$

을 생성합니다.

(참고로 Qiskit은 주로 큐비트 기반 양자 컴퓨터와 함께 사용되도록 설계되었습니다. 따라서 `Statevector`는 2^n 요소를 가진 벡터를 n 큐비트 시스템으로 해석하려고 합니다. 이를 덮어쓰려면 생성자에 `dims` 인자를 전달할 수 있습니다. 예를 들어, `dims=(4,2)`는 시스템이 하나의 4수준 시스템과 하나의 2수준 시스템(큐비트)을 갖고 있다고 Qiskit에 알려줍니다.)

```
W = Statevector([0, 1, 1, 0, 1, 0, 0, 0] / sqrt(3))
W.draw("latex")
```

출력:

$$\frac{\sqrt{3}}{3}|001\rangle + \frac{\sqrt{3}}{3}|010\rangle + \frac{\sqrt{3}}{3}|100\rangle$$

아래 셀은 가장 오른쪽 큐비트(인덱스 0)에 대해 측정을 시뮬레이션합니다. 다른 두 큐비트는 측정되지 않습니다.

```
result, new_sv = W.measure([0])  # measure qubit 0
print(f"Measured: {result}\nState after measurement:")
new_sv.draw("latex")
```

출력:

Measured: 0
State after measurement:

$$\frac{\sqrt{2}}{2}|010\rangle + \frac{\sqrt{2}}{2}|100\rangle$$

셀을 여러 번 실행하여 다른 결과를 확인해보세요. 1을 측정하면 나머지 두 큐비트가 $|0\rangle$에 있다는 것을 알 수 있지만, 0을 측정하면 나머지 두 큐비트가 상태 1/sqrt(2)($|01\rangle + |10\rangle$)에 있다는 것을 알 수 있습니다.

정리

이번 섹션에서는 Qiskit의 `Statevector`와 `Operator` 클래스를 사용하여 양자 시스템을 시뮬레이션하는 방법을 학습했습니다.

1. **텐서곱 (Tensor Product)**
   - `Statevector` 클래스의 `tensor` 메서드를 사용하여 상태 벡터의 텐서 곱을 구할 수 있습니다.
   - 예시: `|0⟩`와 `|1⟩` 상태를 결합하여 `|01⟩`을 생성.
   - `plus`와 `i_state` 상태를 결합하여 새로운 상태 벡터를 생성한 후 출력.
2. **Operator 클래스의 텐서 곱**
   - `Operator` 클래스의 `tensor` 메서드로 두 연산자의 텐서 곱을 구할 수 있습니다.
   - 예시: X 게이트와 I 게이트의 텐서 곱을 계산하여 출력.
3. **상태 벡터의 진화 (Evolve)**
   - 연산자를 사용하여 상태 벡터의 진화를 시뮬레이션할 수 있습니다.
   - 예시: `CX` 연산자를 적용하여 상태 벡터를 변화시킴.
4. **부분 측정**
   - `measure` 메서드를 사용하여 특정 큐비트만 측정할 수 있습니다.
   - 예시: 3 큐비트 상태에서 첫 번째 큐비트를 측정하여 새로운 상태 벡터를 생성.

이러한 기능들을 사용하여 복합적인 양자 시스템과 연산을 다룰 수 있습니다.