Quantum and quantum-inspired linear algebra (Tang, 2023) 학습일지

김태원

최초 작성 : 2023년 9월 4일

최근 편집 : 2023년 9월 4일

제 1 장

블록-인코딩 개요

양자계quantum systems 시뮬레이션은 양자컴퓨터를 만들려는 최초의 동기 가운데 하나였다. 이들 시뮬레이션 가운데 가장 간단한 것은 해밀토니언Hamiltonian 시뮬레이션이다. 해밀토니안 H는 계의 총 에너지에 대응하는 연산자이며, 해밀토니언 시뮬레이션은 1982년 파인만Richard Feynman이 처음 고안한 양자정보학 문제다. 해밀토니안 시뮬레이션이 계의 크기에 따라 지수적인expnential 증가를 보이는 탓에 양자컴퓨터가 필요할지도 모른다는 것이 파인만의 요지였다.

해밀토니언 시뮬레이션.
H가 파울리Pauli 행렬 m개로 구성한 해밀토니언이라고 하자. 이는 다음을 뜻한다.

$$E_a\text{가 파울리 행렬들의 텐서곱일 때, } H = \sum_{a=1}^{m} \lambda_a E_a.$$

e^{-iHt}에 근접하여 $\|U - e^{iHt}\| \leq \epsilon$인 유니타리 U를 구현하는 알고리즘을 찾아라.

여기서 (1) 파울리 행렬, (2) 파울리 행렬들의 텐서곱, (3) e^{-iHt}, (4) 유니타리 같은 용어가 낯설 수 있겠다. 우선 파울리 행렬의 정의에 유니타리의 정의가 필요하므로 유니타리 U부터 정의한다.

정의 1.1. 복소 정방행렬 U는 켤레전치 행렬 $U^\dagger$가 U의 역행렬이기도 할 때 유니타리다. 즉 항등행렬 I에 대해 유니타리 행렬 U는 다음과 같다.

$$U^\dagger U = UU^\dagger = UU^{-1} = I$$

유니타리 행렬 혹은 변환은 어느 유클리드 노름 단위 벡터를 다른 유클리드 노름 단위 벡터로 사상한다. 여기서 '유클리드 노름 단위 벡터'를 큐비트qubit라고 부를 수도 있다.

파울리 행렬은 유니타리 뿐만 아니라 에르미트Hermitian, 거듭involutory 행렬로 정의되므로 에르미트와 거듭 행렬도 살핀다.

정의 1.2. 에르미트 행렬 A는 정방복소행렬로, A의 켤레전치 $A^\dagger$와 A가 같은 행렬이다.

정의 1.3. 거듭행렬은 A가 A 자신의 역행렬인 정방행렬이다. 다시 말해 A가 거듭*involution*인 필요충분조건은 $A^2 = I$다.

정의 1.4. 파울리 행렬은 2×2 복소행렬 세 개의 집합으로, 이들 복소행렬은 에르미트, 유니타리, 거듭 행렬이다.

$$\sigma_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

텐서곱 $\otimes$은 아래처럼 정의된다.

$$x \otimes y \iff (x \otimes y)_{ij} = x_i y_j.$$

그리고 어떤 시간 t에 상태 $|\alpha\rangle$가 주어진다고 하자. 시간이 $t = T$일 때 상태는 $|\alpha(T)\rangle$로 나타내는데 이는 $|\alpha\rangle$에 유니타리 (시간 변화) 연산자 U_1를 적용한 것과 같다. 여기서

$$U_1 = e^{-iHT}$$

H는 해밀토니언을 나타내고, T는 시간을 나타낸다. 반면 해밀토니언 시뮬레이션에서는

$$\|U - e^{iHt}\| \leq \epsilon$$

이므로 구해야 하는 유니타리 U는 해밀토니언 H와 시간 T에 대한 시간 변환 연산 U_i에 근사하는 연산이다. 그리고 해밀토니언 H는 상호작용하는 항들인 E_a의 합으로 볼 수 있고 이때 λ_a는 그런 상호작용의 강도를 결정한다.

보통 이 문제의 답은 충분히 큰 r에 대해 아래와 같은 트로터Trotter 근사로 주어진다.

$$e^{-iHt} \approx (e^{-iE_1 t/r} e^{-iE_2 t/r} \dots e^{-iE_m t/r})^r.$$

하지만 이 해는 최적과는 거리가 멀다. 왜냐하면 위와 같은 근사는 양자회로의 게이트 개수에 대응하는 게이트 복잡도gate complexity가 $\mathrm{poly}(1/\epsilon)$만큼 요구되기 때문이다. 다시 말해 ϵ의 역원에 대해 다항인 만큼 게이트들이 요구되기 때문이다. 개선된 알고리즘들은 저자Ewin Tang가 제시하는 프레임워크을 따른다. 이 프레임워크는 다음과 같은 절차를 지닌다.

(i) "블록-인코딩block-encoding"이라는 양자 회로의 유형을 정의한다.

(ii) 주어진 λ_a, E_a에 대해 H의 효율적인 블록-인코딩을 구성할 수 있다는 사실을 보인다.

(iii) H에 대한 블록-인코딩을 적게 사용하여 e^{-iHt}의 근사에 대한 블록-인코딩을 취할

수 있다는 사실을 보인다.

(iv) 이 블록-인코딩을 사용해 한 상태에 대해 근사를 내놓는다.

1.1 블록-인코딩

정의 1.5. 주어진 $A \in \mathbb{C}^{r\times c}$에 대해 $U \in \mathbb{C}^{d\times d}$는 U가 $\mathcal{O}(Q)$ 게이트로 구현가능하고 아래 조건을

$$B_{L,1}^{\dagger} U B_{R,1} = A$$

항등 행렬의 첫 r과 c열인 $B_{L,1} \in \mathbb{C}^{d\times r}, B_{R,1} \in \mathbb{C}^{d\times c}$에 대해 만족하면 A의 Q-블록 인코딩이라고 한다.

다시 말해, U의 임의의 원소 $\cdot$ 에 대해

$$U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}$$

인 경우다. 또한 $\prod_L = B_{L,1} B_{L,1}^{\dagger}, \prod_R = B_{R,1} B_{R,1}^{\dagger}$이 각각 $B_{L,1}$과 $B_{R,1}$의 생성에 대한 사영에 대응하도록 한다.

d, r, c를 2의 지수승으로 자주 나타낼 예정인데 모든 것을 큐비트로 다루려는 목적이다. 이에 정의 1.5는 아래처럼 나타낼 수 있다.

$$(\langle 0|^{\otimes aL} \otimes I) U (|0\rangle^{\otimes aR} \otimes I) = A.$$