

기계학습을 통한 양자 회로 에러 보정 방법

김창준, 박경덕, 이준구*
한국과학기술원 전기 및 전자공학부

changjun0605@kaist.ac.kr, dkp.quantum@gmail.com, *rhee.jk@kaist.edu

Quantum Circuit Error Mitigation Via Machine Learning

Changjun Kim, Daniel K.Park, June-Koo Kevin Rhee
School of Electrical Engineering, KAIST

요 약

본 논문은 두 가지 방법의 기계학습을 통한 양자 회로 에러 보정 기술을 제안한다. 여러 환경의 에러 시뮬레이션과 실제 IBM 머신의 실험 결과를 이용하여 결과를 생성하고, 수학적 모델과 일반적인 DNN(Deep Neural Network) 및 Concatenated DNN 을 통해 이를 보정 및 분석했다. 제안한 방식은 2 큐비트 시스템의 길이 1-10 의 임의의 회로 관측 결과를 바탕으로 학습한 후, 다른 임의의 길이 1-20 의 임의의 회로를 보정하고, 유효한 결과를 얻을 수 있으며, 실제와 비슷한 환경에서 가장 좋은 결과를 보인다. 이 방법은 추가 큐비트를 필요로 하지 않으며 NISQ (Noisy Intermediate-Scale Quantum) 회로에서 시간적 이득을 얻을 수 있다.

I. 서 론

양자 컴퓨팅기술은 무어의 법칙의 한계를 맞고 있는 고전 컴퓨팅 방식의 새로운 돌파구로서 주목받고 있다. 하지만 현재 양자 하드웨어의 에러로 인해 원하는 성능을 내지 못하는 경우가 많다. 따라서 양자 에러 정정 혹은 양자 에러 보정이 필요하다. 하지만 양자 에러 정정(Quantum Error Correction)의 경우 아주 많은 양의 큐비트를 추가로 필요로 한다. [1] 큐비트 개수에 따라 자원 사용량이 크게 증가하기 때문에, 추가로 큐비트를 사용하지 않는 에러 보정 방법(Quantum Error Mitigation)이 제안되었다. [2, 3] 본 논문에서는 기계학습을 통하여 양자 회로에서 발생하는 computational 기저별 관측 확률 값의 에러를 보정하고자 한다.

II. 에러 모델

본논문에서는 depolarizing 에러 모델과 IBM Q 머신의 실제 에러 모델을 반영해 시뮬레이션을 진행하고, IBM Q 머신에서 실제 회로를 실행한 결과를 사용하였다.

2.1. Depolarizing 에러 모델

Depolarizing 에러는 아래와 같이 Kraus representation 을 사용하여 설명할 수 있다.

$$\rho_{\varepsilon} = (1 - \varepsilon)\rho + \frac{\varepsilon}{4^N - 1} E \rho E^{\dagger}$$

$$\text{where, } E = \sum \bigotimes_{i=1}^N \{I, X, Y, Z\} / \bigotimes_{i=1}^N \{I\}$$

한 큐비트 게이트의 에러율 ε_1 을 10^{-3} 으로, 두 큐비트 게이트의 에러율 ε_2 를 10^{-2} 으로 설정하였다. 또한 관측의 경우 $a, b \in \{0, 1\}$ 인 $M_{ab} = |ab\rangle\langle ab|$ 를 사용하여 $\bigotimes \{I, Z\}$ 로 구성된 양자 밀도 행렬의 대각 값만 사용하였다.

두 큐비트로 이루어진 양자 밀도 행렬은 아래 식과 같이 표현할 수 있다.

$$\rho = \frac{I \otimes I}{4} + \sum_{i,j \neq (1,1)}^4 \gamma_{i,j} \sigma_i \otimes \sigma_j, \text{ where } \sigma_i, \sigma_j \in \{I, X, Y, Z\}$$

$M_{ab} = |ab\rangle\langle ab|$ 로 관측하게 될 경우 위 식에서 필요한 값들은 $\gamma_{1,4}, \gamma_{4,1}, \gamma_{4,4}$ 이고, depolarizing 에러를 지날 경우 다음과 같이 변하게 된다.

$$\begin{aligned} \tilde{\gamma}_{1,4} &= \left(1 - \frac{4}{3}\varepsilon_1\right)^{N_{12}} \left(1 - \frac{16}{15}\varepsilon_2\right)^{N_2} \gamma_{1,4} \\ \tilde{\gamma}_{4,1} &= \left(1 - \frac{4}{3}\varepsilon_1\right)^{N_{11}} \left(1 - \frac{16}{15}\varepsilon_2\right)^{N_2} \gamma_{4,1} \\ \tilde{\gamma}_{4,4} &= \left(1 - \frac{4}{3}\varepsilon_1\right)^{N_{11}+N_{12}} \left(1 - \frac{16}{15}\varepsilon_2\right)^{N_2} \gamma_{4,4} \end{aligned}$$

이 때, N_{11}, N_{12} 는 큐비트 1 과 큐비트 2 에 걸리는 한 큐비트 게이트의 개수이고, N_2 는 두 큐비트 게이트의 개수다. 이를 이용하여 관측할 경우 각 상태의 관측 확률은 다음과 같다.

$$P(ab) = \frac{1}{4} + (-1)^b \tilde{\gamma}_{1,4} + (-1)^a \tilde{\gamma}_{4,1} + (-1)^{a+b} \tilde{\gamma}_{4,4}$$

2.2 IBM Q 시뮬레이션

IBM Q 에서 모델링하여 제공하는 에러 모델을 불러와 시뮬레이션을 수행하였다. Depolarizing 에러, thermal relaxation 에러, readout 에러 등을 포함하는 모델이고, 머신은 초전도 트랜스몬 큐비트를 사용하는 5 큐비트 기기, ibmq_5_yorktown 의 2020 년 6 월 25 일의 정보를 불러왔다. 이 때 한 큐비트 게이트 에러의 경우 큐비트 0 과 큐비트 1 에 대해 각각 6.25×10^{-4} , 6.70×10^{-4} 의 에러율을 가지고, 두 큐비트 게이트 에러의 경우 1.65×10^{-2} 의 에러율을 가졌다. 이 때, readout 에러 보정을 수행하지 않은 경우와 수행한 경우에 대해 분석하였다.

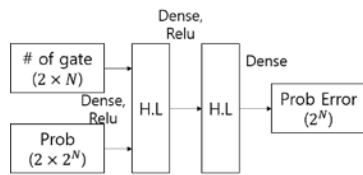


그림 1. 기계 학습 모델 (DNN)

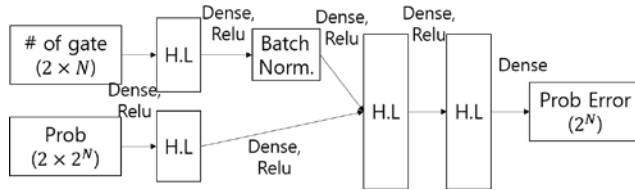


그림 2. 기계 학습 모델 (Concatenated DNN)

2.3 IBM Q 실험

실제 IBM Q 머신에서 같은 회로를 실행 및 관측하여 결과를 분석하였다. 이 때 머신은 2.2 와 같은 특징을 가졌다. 2.2 의 시뮬레이션과 2.3 의 실험은 다른 결과를 가질 수 있기 때문에 [4], 실험도 따로 진행했다.

2.4 데이터 생성

회로의 경우 S, T, H 의 한 큐비트 게이트와 CZ 의 두 큐비트 게이트를 가지도록 했다. 임의의 게이트를 가진 길이 1 부터 20 까지의 회로를 설계했고, 각 길이별로 400 개의 회로를 설계했다. 각 게이트를 지나기 전과 지난 후의 결과를 관측하여 데이터로 만들었다.

각 회로의 입력으로 각 큐비트는 0 과 1 만을 가지는 모든 계산 기저를 사용했다. 2.1-2.3 에서 사용한 방법을 이용해 4 가지 데이터를 생성하고, 에러가 없는 데이터를 추가로 생성해 총 5 가지의 데이터를 사용했다.

III. 에러 보정 방법

2 단에서 사용한 depolarizing 에러를 역 연산하여 에러를 보정하는 방법과 기계 학습을 사용하여 에러를 보정하는 두 가지 방법이 사용되었다.

3.1. Depolarizing 에러 보정

2.1 을 역 연산하면 다음과 같은 꼴로 에러를 보정할 수 있다.

$$P(ab) = \frac{1}{4} + \frac{(-1)^b \tilde{\gamma}_{1,4}}{E_1} + \frac{(-1)^a \tilde{\gamma}_{4,1}}{E_2} + \frac{(-1)^{a+b} \tilde{\gamma}_{4,4}}{E_3}$$

$$E_1 = \left(1 - \frac{4}{3} \varepsilon_1\right)^{N_{12}} \left(1 - \frac{16}{15} \varepsilon_2\right)^{N_2}$$

$$E_2 = \left(1 - \frac{4}{3} \varepsilon_1\right)^{N_{11}} \left(1 - \frac{16}{15} \varepsilon_2\right)^{N_2}$$

$$E_3 = \left(1 - \frac{4}{3} \varepsilon_1\right)^{N_1+N_{12}} \left(1 - \frac{16}{15} \varepsilon_2\right)^{N_2}$$

3.2 기계학습을 통한 에러 보정

일반적인 deep neural network(DNN)를 이용한 방법과 첫번째 레이어를 concatenate 한 concatenated deep neural network(Concatenated DNN)의 두 가지 방법이 기계학습으로 사용되었다.

기계학습의 입력으로 첫 번째 게이트를 지나기 전의 관측 확률 (2^N 개), 특정 게이트를 지난 후의 관측 확률 (2^N 개), 각 큐비트가 지난 한 큐비트 게이트 개수 (N 개), 두 큐비트 게이트 개수 (N 개)를 넣어주어, 총 $2N + 2^N$ 의 입력을 가진다. 기계 학습의 출력으로 특정 게이트를 지난 후의 확률 에러량 (2^N 개)을 갖게 된다.

Error Model	None	DNN	C.DNN	Depol.
Depolar.	2.5e-4 (1)	8.0e-5 (0.32)	7.0e-5 (0.28)	2.7e-5 (0.11)
Simulation	4.2e-4 (1)	1.6e-4 (0.37)	1.5e-4 (0.35)	3.0e-4 (0.70)
Sim. w/ Mit.	2.1e-4 (1)	1.6e-4 (0.75)	1.3e-4 (0.63)	2.1e-4 (0.89)
Real Model	1.8e-3 (1)	1.1e-3 (0.63)	9.7e-4 (0.54)	1.8e-3 (0.97)

표 1. 에러 보정 결과

Concatenated DNN 의 경우 게이트 개수가 회로의 길이가 늘어남에 따라 늘어나기 때문에, 이를 정규화 해주기 위해 첫 번째 히든 레이어를 지난 후 batch normalization 을 수행 해 주었다.

IV. 결과

각 수정에 대해 에러 보정 결과에 대해 표 1 과 같은 결과를 갖는다. 4 가지 종류의 에러를 적용했을 때 확률과 에러가 없을 때의 확률 차의 RMSE(Root Mean Squared Error)를 결과로 갖는다.

$$RMSE(P_\varepsilon) = \sqrt{\frac{(P(ab) - P_{\varepsilon,mut}(ab))^2}{2}}$$

또한 에러 보정을 하지 않은 에러 확률 크기를 기준으로 다른 에러 보정 방법의 에러량의 비율을 괄호안에 표기하였다.

Depolarizing 에러 보정 방법을 사용한 경우 depolarizing 에러가 있는 경우 큰 보정 값을 가지지만 다른 방법의 경우 depolarizing 에러 외에 다른 에러도 존재하기 때문에, 큰 에러 보정 값을 가지지 못한다.

Concatenated DNN 의 경우 실제 환경과 비슷한 세 가지 에러 모델에서 가장 좋은 성능을 보였으며, depolarizing error 의 경우에도 에러가 72%가 줄어드는 큰 이득을 보인다.

ACKNOWLEDGMENT

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터지원사업의 연구결과로 수행되었음" (IITP-2020-2018-0-01402)

한국연구재단의 지원을 받아 수행된 기초연구사업 (NRF-2019R1I1A1A01050161)

참 고 문 헌

- [1] Shor, Peter W. "Scheme for reducing decoherence in quantum computer memory." Physical review A 52.4 (1995): R2493.
- [2] Endo, Suguru, Simon C. Benjamin, and Ying Li. "Practical quantum error mitigation for near-future applications." Physical Review X 8.3 (2018): 031027.
- [3] Temme, Kristan, Sergey Bravyi, and Jay M. Gambetta. "Error mitigation for short-depth quantum circuits." Physical review letters 119.18 (2017): 180509.
- [4] Blank, Carsten, et al. "Quantum classifier with tailored quantum kernel." npj Quantum Information 6.1 (2020): 1-7.