Quantum neural distinguisher for speck 32/64

https://youtu.be/GwB-Qkz9u6M

HSU 한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

내용들의 출처..

- 논문들
  - https://arxiv.org/pdf/2109.11676.pdf
  - https://arxiv.org/pdf/1905.10876.pdf
  - https://arxiv.org/pdf/2203.01340.pdf
  - https://physics.aps.org/featured-article-pdf/10.1103/PhysRevX.10.041038

- 페니레인 디스커션 포럼
  - https://discuss.pennylane.ai/

- 구글링

1. NISQ

- **Noisy intermediate-scale quantum era**
  - 중간 규모의 양자컴퓨터이지만, 오류가 발생
  - 오류 정정이 가능한 정도의 규모는 아님
  - 복잡한 특정 작업에 양자 신경망을 하이브리드 방식으로 적용하는 접근 방식이 대다수

- **양자컴퓨터를 사용할 수는 있지만, 고전적인 방법을 능가할 만큼 강력하지 않음**
  - 고전 신경망보다 더 나을 수도 있고, 아닐 수도 있음
  - 양자 신경망의 목표 중 하나는 양자 신경망이 유리한 특정 상황을 식별하는 것
    → 양자 컴퓨터가 이점을 제공할 수 있는 작업이 존재

- **양자 이점만이 양자 인공지능의 올바른 방향은 아니라는 견해들이 있음**
  - 양자가 고전 인공지능을 능가할 수 있는 것은 매우 작은 규모에서만 가능
  - 이러한 작은 규모의 실험을 확장한다고 해서 그 작업도 양자 이점이 있을 거라는 보장은 없음

2. Qubit

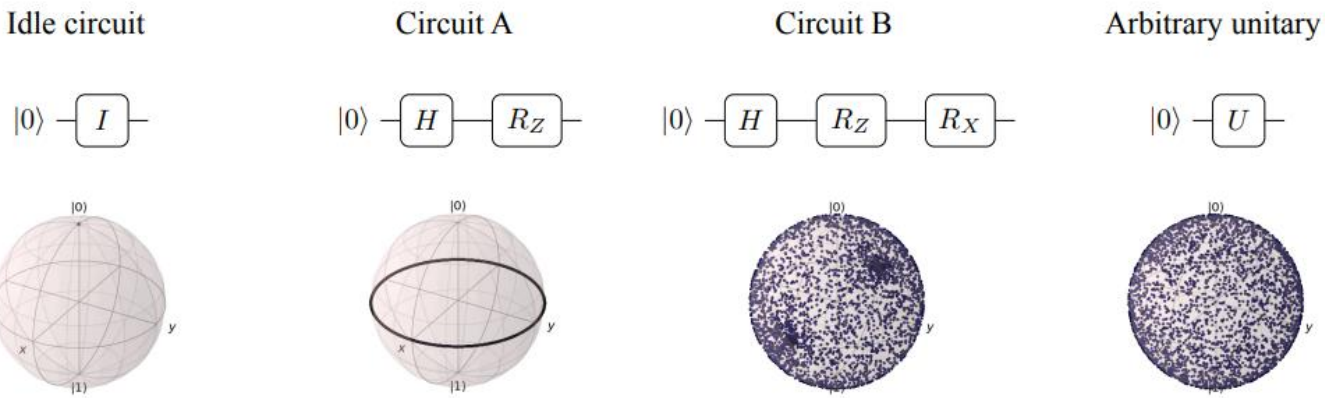
- **현재 양자 인공지능에 사용 가능한 최대 큐비트는 대략 16 큐비트**
  - 지금까지 본 연구들에서는 2~10 큐비트 정도 사용하는 것 같음

- **큐비트의 수가 증가함에 따라 시뮬레이션 난이도가 기하급수적으로 증가**
  - 필요한 컴퓨팅 파워와 소요 시간이 증가
  - 시뮬레이터에서는 큐비트가 증가한다고 해서 노이즈가 생기는 건 아님 (이런 의미의 난이도는 아님)

- **'양자 회로의 폭 (큐비트)가 증가하면 더 나은 결과를 얻을 수 있다'라는 주장은 합리적이나, 시뮬레이션이 매우 어려움**
  - 현재로서는 성능 향상을 위해 큐비트 수를 늘리는 것만이 좋은 접근 방법은 아닌 것 같음
  - 여러 게이트를 조합해서 많은 데이터를 효과적으로 임베딩하거나, 데이터 재업로딩, 회로 자르기 등의 기능을 사용

3. Gate

- **기저 축이 다른 회전 게이트 (Rx, Ry, Rz)를 조합할수록 표현력이 높아짐**
  - H-Rz : z축을 기준으로 회전할 때 가질 수 있는 값들을 모두 가지게 됨
  - H-Rz-Rx : H-Rz의 점들이 x축 기준 회전하면서 가질 수 있는 값들을 가짐



- **Hadamard**
  - Hadamard 게이트가 사용되는 경우 존재
  - 회전 게이트가 중첩을 만들 수 있음 → 회전을 통해 큐비트 구 위의 값을 만들 수 있음
  - 양자 신경망에서는 회전 게이트를 주로 사용하기 때문에 일반적인 양자 회로보다 Hadamard 게이트가 덜 사용됨

4. Embedding

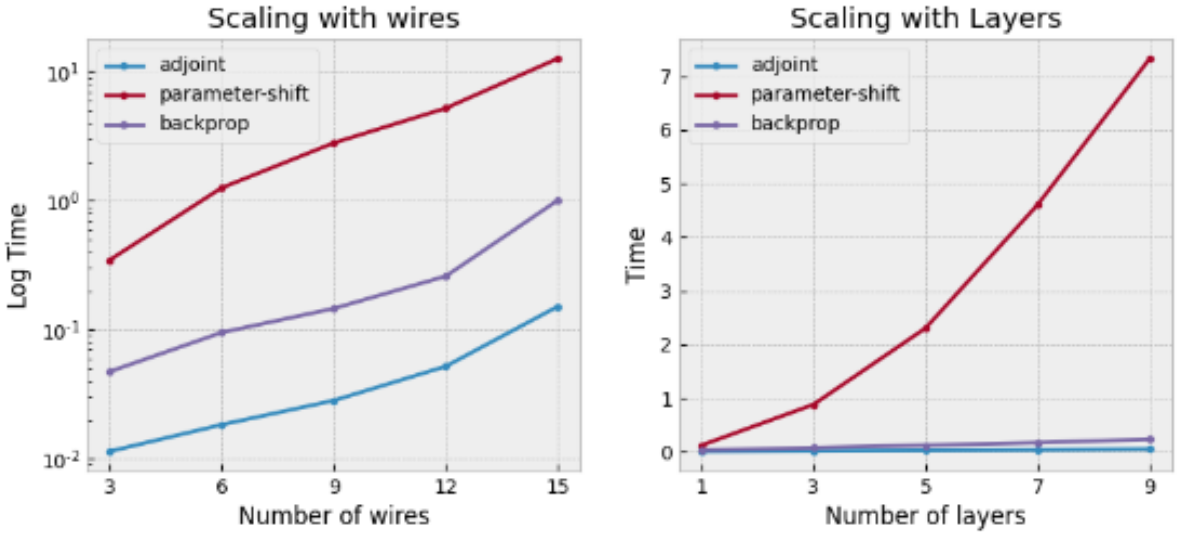
- **각 / 진폭 / 기저 임베딩 등 여러 종류의 임베딩 기술 존재**
- **데이터의 종류와 문제에 따라 선택**

  - **진폭 임베딩**
    - n개의 큐비트로 2^n개의 데이터 임베딩이 가능
    - 필요 큐비트가 적지만 depth가 높고 비교적 복잡
    - 실수형 데이터에 적합 (실수 데이터 또는 가중치)

  - **기저 임베딩**
    - n개의 큐비트로 n개의 데이터 임베딩이 가능
    - 고전 데이터 $1,0,1$을 양자 상태 $|101\rangle$로 그대로 임베딩 → 이진 데이터에 적합
    - 간단하지만 많은 큐비트가 필요

  - **진폭 임베딩은 많은 데이터 포인트를 단일 회로에 임베딩 할 수 있으나, 그만큼 해석이 어려워짐**
    - Speck 32/64 데이터는 진폭 임베딩을 사용한 경우보다 기저 임베딩을 사용한 경우가 더 높은 정확도를 얻었음

5. 시뮬레이터 및 미분 방식



- **페니레인 시뮬레이터**
  - 기본 / C++ 기반 가속화 시뮬레이터 / 노이즈 시뮬레이터

- **미분 방식**
  - **Adjoint**
    - 메모리 감소 가능 → 더 많은 큐비트 시뮬레이션에 도움 (작년에는 에러났었는데 지금은 됨)
    - 가속화 시뮬레이터와 함께 사용하여 속도 향상 가능
    - 노이즈 시뮬레이터에서는 지원하지 않음
  - **Backprop**
    - 학습 소요 시간이 매우 큼 (adjoint의 약 두 배 정도였음)
    - 노이즈 시뮬레이터에서 지원은 가능하지만, 메모리 사용량이 매우 커서 OOM (Out Of Memory) 발생
  - **Parameter shift**
    - 레이어와 큐비트가 늘어날수록 소요 시간이 매우 크게 증가
    - 파라미터 하나 당 최소 2회 연산되어야 하며, 이는 시뮬레이터에서 특히 오래 걸림
    - 실제 양자 하드웨어에서 적합한 방식

6. 파라미터와 회로의 복잡도

- **회로의 표현력과 복잡도**
  - 큐비트의 수나 양자 레이어의 수가 증가 → 그에 따라 게이트의 수도 증가
  - 여러 회전 게이트와 얽힘이 추가될수록 회로의 연결성과 표현력이 증가
  - 특히, 2 큐비트 게이트 (CRx, CNOT 등)가 회로의 연결성을 결정
    → 그러나 해당 게이트들이 복잡성 증가의 원인이 됨 → 회로의 깊이와 게이트 수에서 복잡도 증가
  - 일반적으로 적은 depth에서 많이 얽힐수록 (연결성이 높을수록) 효율적인 표현 공간을 가질 수 있음
    → 이러한 레이어들이 추가될수록 표현력이 높아지지만, 복잡성이 높아질 수 있으므로 주의
- **하고자 하는 작업에 비해 회로의 복잡성이 높으면 과적합 발생**
  - 또한 회로마다 레이어 수, 게이트 수 및 종류 등에 따른 표현력이 다르므로 무조건 파라미터가 많을수록 좋은 것은 아님
  - 파라미터가 많으면 회로 최적화 난이도가 높아짐
- **현재 NISQ에서는 큐비트의 수나 회로의 깊이, 연결성에 제한이 있으므로 적절한 회로 구성 필요**
  - 큐비트를 많이 사용한다고 무조건 좋은 결과가 나오지는 않음
  - 모든 큐비트가 얽히는 것이 좋다고 알려져있으나 비용 (회로 깊이, 파라미터 수)이 매우 비쌈
  - 과파라미터화 된 회로의 기준이 4 큐비트에 25개의 파라미터라고 제안한 연구 결과 존재
  - 실제로 실험해보니
    - 4 큐비트에 60개의 파라미터인 경우: 3~4번 정도 돌리면 2번은 과적합
    - 4 큐비트에 20개의 파라미터인 경우: 2번 돌렸을 때 과적합 없음 (한 번 더 실행 중)

7. 평가 방식

- **문제와 데이터에 따라 다 다르기 때문에 정확도를 제외하면 평가 기준이 애매한 것 같음**
- **성능 (정확도)은 앞서 살펴본 매개변수화 된 회로의 연결성과 표현력에 따라 달라짐**
  - 정확도, 회로의 2 큐비트 게이트의 수, 회전 게이트의 수와 종류, 파라미터의 수
    → 해당 요소들이 연결성과 표현력에 영향을 주므로 이를 중점으로 평가하면 될 것 같음

- **현재 상황에서는 고전 신경망보다 이점을 얻기는 어려움**
  - 지금의 시뮬레이터로는 많은 데이터를 학습하기 어려움
  - 그러나, 데이터 수에 비해 데이터의 복잡도가 낮지 않음 (64개의 데이터 포인트)
  - 고전 신경망에서는 7 라운드 이상 분석 가능하지만, 이를 위해서는 10^7 개 이상의 데이터가 필요함
    → 양자 신경망에서는 5 라운드 분석에 1만 개 정도의 데이터를 사용함
  - 다음과 같이 3 가지를 비교하고자 함 (아직 생각 중..)
    - 1: 데이터 및 파라미터 제한 없는 고전 신경망을 사용할 경우의 5 라운드에 대한 성능 (Gohr 논문)
    - 2: 데이터 수 1만 개로 제한, 최소한의 고전 레이어 (입력층 - 은닉층 1개 - 출력층)를 사용한 고전 신경망의 5 라운드 성능
    - 3: 양자 신경망 + 2을 사용한 5 라운드 성능 → 양자 신경망 + 최소한의 고전 레이어를 했을 때의 5 라운드 distinguisher 성능

- **NISQ 장치에서 사용하기 적합한 회로인지**
  - 시간, 큐비트 수, 등등... 다른 기준이 있는지 보겠습니다.

8. Quantum Neural Distinguisher for Speck 32/64

Amplitude/Adjoint	Tr	Val	Ts	
6-qubit	0.53	0.51	0.5	거의 학습되지 않는 수준 가중치가 아닌 데이터 자체를 임베딩 하고 싶어서 적합하지 않음

Basis/Adjoint	Tr	Val	Ts	15 레이어 기준 (과파라미터로 추정)
8-qubit	0.66~0.72	0.49~0.7	0.49~0.72	편차가 큼 (과적합 날 때도 있고 아닐 때도 있음), 그래도 16 큐비트에 비해 정확도는 높음 → 큐비트의 수에 따라 얽힘이 달라지는데, 이게 영향이 있는 것 같음
16-qubit	0.62	0.49	0.47	과적합/ Tr도 낮음

과파라미터의 기준을 며칠 전에 알게 되어서 16 큐비트도 레이어 수를 줄여서 돌려보고 있습니다.
(1 epoch에 3600초 정도 밖에 안 걸려서 실험 한 번 당 10시간 정도라서 3번 돌려보려면 2~3일 안에 나올 것 같습니다.)

Basis/Adjoint	Tr	Val	Ts	6 레이어로 줄임
8-qubit	0.74	0.72	0.77	편차 감소, 정확도 비슷 양자로도 가능하지만 양자를 추가함으로써 이득이 딱히 없음

Classical	Tr	Val	Ts	
1 prediction head	0.79	0.75	0.76	제한 없는 경우 7 라운드 이상까지 되지만 양자 추가한 것과 비교하기 위해 데이터와 모델 제한

양자 인공지능으로도 가능하지만 지금의 양자 인공지능으로는 적합한 작업은 아닌 것으로 생각됨
→ 그러나 계속 개발되고 있어서 다른 방법들이 더 있을 수도 있음

감사합니다.