Quantum Neural Network와 암호 분석

https://youtu.be/IS085KkjBEU

Quantum Computer

Quantum Neural Network

Quantum Neural Network based Cryptanalysis

*QNN 세미나에서 했던 거에 추가된 내용으로 구성했습니다..

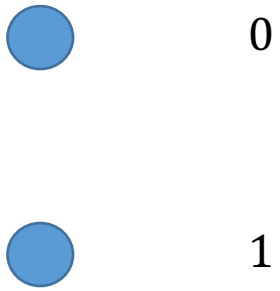
https://youtu.be/XEsoJ9zGcTY

Quantum Computer

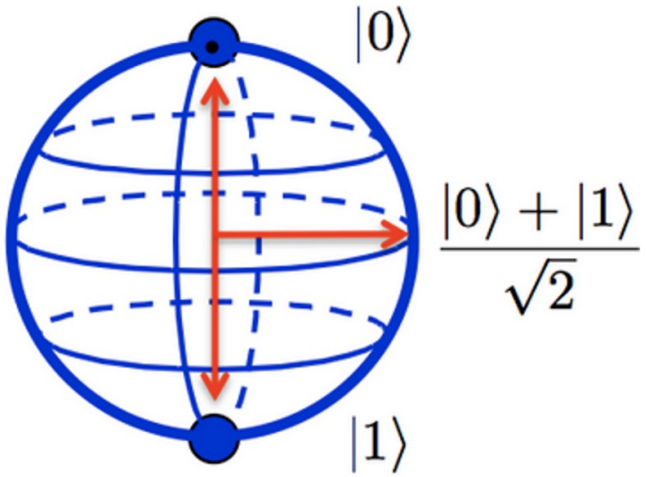
- **양자 역학적인 현상을 기반**으로 연산하는 컴퓨터
- **큐비트 및 게이트를 통해 양자회로를 구성**하고 이를 통해 연산
- 현재 IBM에서 **127큐비트** 달성 (Eagle 프로세서)
- 이후에는 **Osprey** (433 큐비트) - **Condor** (1121 큐비트) 개발 예정
- 양자 컴퓨터는 연산 시 발생하는 **오류로 인해 연산 정확도가 감소**할 수 있음
  하나의 논리적인 큐비트 (오류가 발생하지 않는 큐비트)를 위해서는 수십개의 물리적인 큐비트가 필요하며,
  오류 정정 기술 또한 필요
- **NISQ 시대**
  현재는 중간 규모의 양자 컴퓨터 → 오류 수정에 사용할 큐비트가 충분하지 않음 → noise 존재
  그러나, 양자 우위를 보여주기에는 충분 (고전 컴퓨터의 계산 능력을 뛰어넘음)

Qubit

- 양자역학의 **중첩, 측정, 얽힘**과 같은 개념에 기반
- 양자 컴퓨터에서는 기존 컴퓨터의 비트와 같이 **큐비트**를 사용
  - 기존의 비트는 0 or 1을 가짐
  - 큐비트는 **0과 1을 확률로써 동시에 가질 수 있음 (중첩상태)**
- 측정 시 하나의 값으로 결정



Bit



Qubit

Qubit

- **측정하기 전은 0과 1의 선형 조합, 측정 후 하나의 값으로 결정 (0 또는 1로 감)**
  - $|\psi\rangle =$ 임의의 큐비트 상태 $= \alpha|0\rangle + \beta|1\rangle$ ($\alpha^2 + \beta^2 = 1$, 확률)

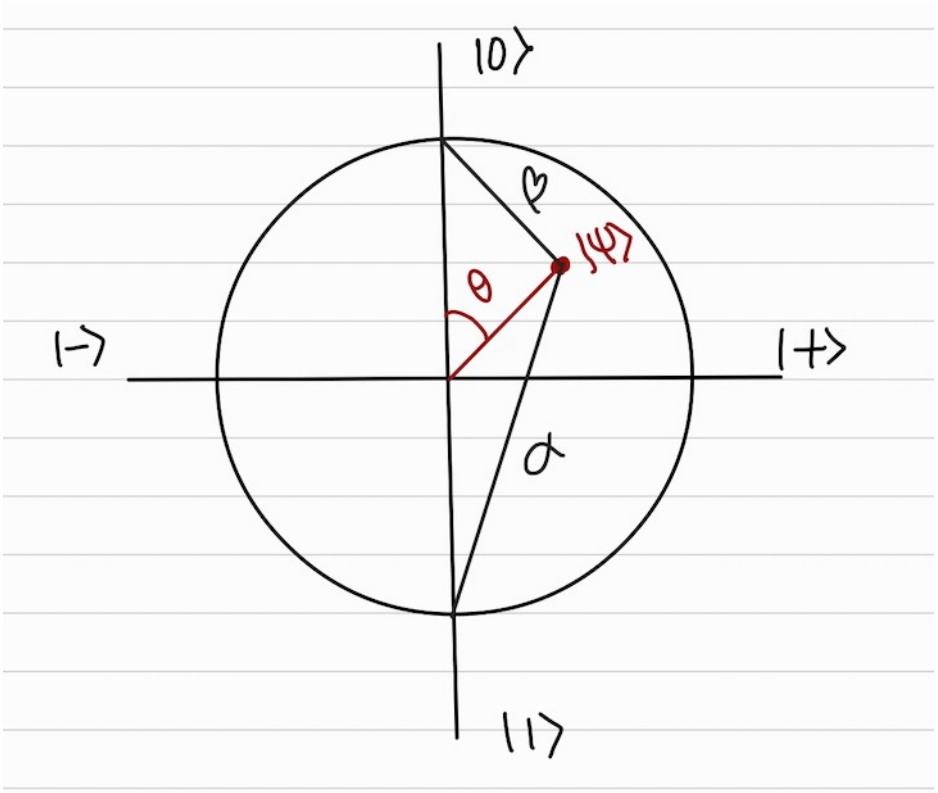
  → 이와 같이 0과 1이 확률로 존재 (α^2, β^2는 0과 1이 될 확률)

- **양자 상태는 벡터이고, 행렬 곱을 통해 상태 변경 가능**

  → 행렬 곱을 통해 큐비트 상태를 또 다른 벡터로 변경

  → 양자 게이트도 행렬 곱 통해 큐비트 상태를 변화시키는 것

  (단위 행렬(I)은 상태변화 시키지 않음)



Quantum Gate

- **다음과 같은 게이트들이 주로 사용**

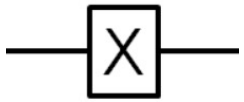
1 또는 0이 될 확률을 바꾸는 것이 아니라 상태를 바꿈

- **Hadamard gate (H)**

  Qubit의 초기상태에서 0과 1의 상태를 동시에 가질 수 있도록, 중첩 시킴
  해당 게이트를 두 번 거칠 경우 원래 상태로 돌아옴

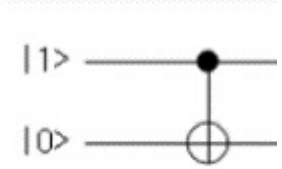
- **X gate**

  큐비트의 상태를 0이었으면 1로, 1이었으면 0으로 변경
  중첩 상태의 큐비트의 경우 0과 1이 될 수 있는 확률을 변경



- **CNOT gate**

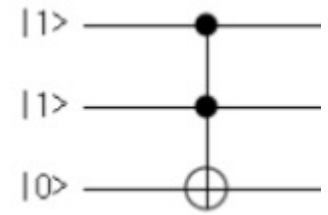
  한 큐비트가 다른 큐비트에 영향을 끼치는 얽힘 상태 관찰 가능
  첫 번째 큐비트가 1인 경우에 두 번째 큐비트에 NOT 게이트 연산



첫 큐비트가 1이므로
두번째 큐비트의 0값이 1로 반전

- **Toffoli gate**

  앞의 두 비트가 모두 1인 경우 세 번째 큐비트에 NOT 연산



위의 두 큐비트가1이므로
세번째 큐비트의 0값이1로 반전

- **SWAP gate**

  두 큐비트의 상태를 서로 바꾸는 게이트

Quantum Gate

- **Rotation**

  $|0\rangle$ 을 θ만큼 시계방향으로 회전시켜 $|\psi\rangle$ 로 바꾸는 것

  → 회전을 통한 상태 변화

- **Ry** → y축 중심 회전 (Rx면 x축 중심 회전)

  오른쪽 행렬은 y축 중심 회전 행렬

  → 이 때의 ***θ*는 매개변수**

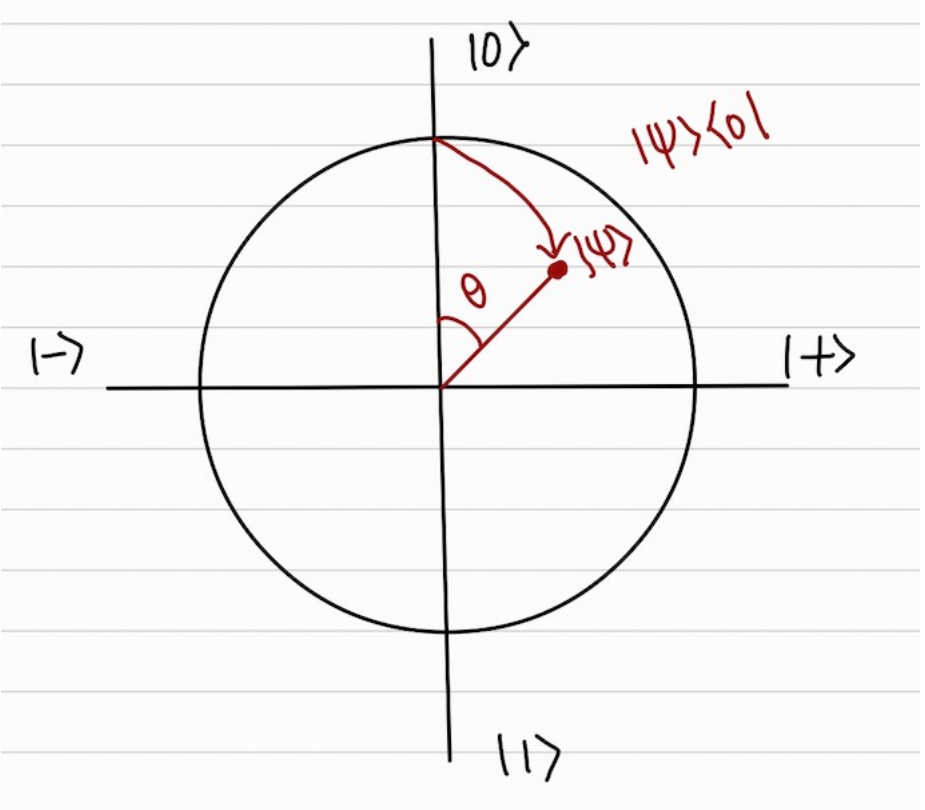
$$\begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

- **Controlled Ry → 결합 확률**

  다른 큐비트와 얽힘 상태로 만든 후, 다른 확률과의 결합 확률 계산

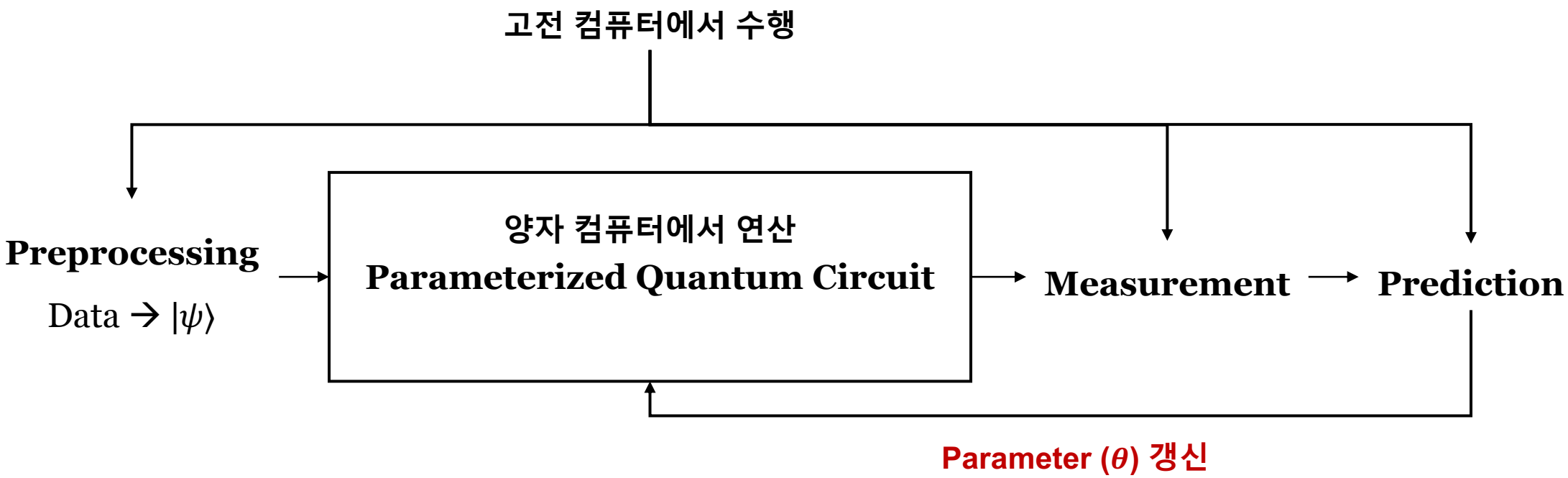
  → CNOT과 비슷하게 작용 (제어 큐비트가 1일 경우, X 대신 Ry 적용)

  → **cx - ry - cx 순으로 적용하여 회전 후, 얽힘을 풀어줌**



* $\frac{\theta}{2}$ 인 이유는 서로 반대인 $|0\rangle$ 과 $|1\rangle$ 을 기저로 하여 큐비트의 상태를 나타내기 때문

Quantum Neural Network



*Classical-quantum hybrid network → 양자회로 뒤에 classical NN 붙임

Data

- 애초에 **양자 데이터를 생성**하거나 Classical computer에서의 데이터를 **Quantum data로 인코딩**

  1. 어떤 큐비트 상태에 대해 회전연산 적용하여 불확실성 주어 quantum data 생성 가능

  2. Classic data → quantum data 과정을 말함 ($\text{Classic data} \rightarrow |\psi\rangle$)

     → Hadamard gate 및 데이터를 입력한 후, 회전연산

Parameterized Quantum Circuit - θ

- 양자 회로의 매개변수를 바꾸는 것 → 큐비트를 θ씩 회전시키면서 회전각 θ를 바꿈 → 큐비트 구면 위의 다른 점이 됨 (다른 상태를 가짐)
  → 즉, **θ (신경망의 매개변수인 가중치)를 변경해가면서 적절한 값(적절한 가중치)을 찾아냄**
- 0또는 1로 측정될 확률에 따라 θ를 조정 → 기존 신경망에서도 결과 값에 따른 loss 계산 후, 가중치 조정
- **파라미터화 된 양자회로를 통해 측정 확률 조정 및 매개변수 제어가 가능**
  → QNN 학습 과정

Parameterized Quantum Circuit와 기존 신경망

- **행렬 곱 연산이 기존 신경망의 주 연산**

  → 큐비트 상태 또한 **행렬 곱(양자 게이트)을 통해 변경**

    → 코드 여러 개를 봤을 때 h, rx, ry, rz, cx 정도 주로 사용

- **기존 신경망의 활성화 함수는 비선형 연산**

  → 양자회로 내부 게이트를 **비선형 연산 포함하여 구성**

- **기존 신경망에서 최적화 함수 사용**

  → 양자 신경망 또한 **최적화 함수 사용**

    1. 하이브리드 신경망의 경우 기존 신경망의 최적화 함수 사용 가능 → tensorflow quantum 등은 양자회로를 레이어로 사용 가능
    2. COBYLA, SPSA, SLSQP 와 같은 최적화 알고리즘이 있음 (QSVM의 경우 SPSA 사용)

Parameterized Quantum Circuit을 활용한 암호분석

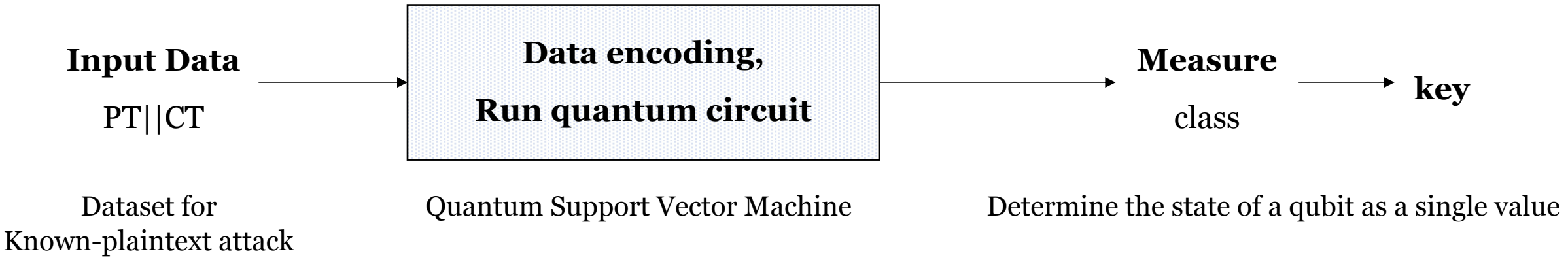
Parameterized Quantum Circuit을 활용한 암호분석

- **현재 거의 연구되지 않고 있음**
- 최근에 제가 **Caesar 암호를 QSVM으로 알려진 평문 공격** 수행했는데

  1. QSVM 말고 tf.quantum 등의 다른 라이브러리로 qnn 구성

  2. 양자 회로 부분 보완 (reuploading 기법 적용해서 큐비트 재사용이나 실제 양자 컴퓨터 사용, 또는 회로 내부 구조 변경)해서

  다른 암호들도 분석이 가능한지 해볼 생각입니다..



Quantum Support Vector Machine을 활용한 암호분석

- **알려진 평문 공격**

  → 평문 및 암호문 쌍(data)으로 키(label) 찾기

  → 평문 및 암호문은 비트로 표현, 키는 십진수로 하여 label 지정 (지도학습) → 4차원 데이터면, 4개의 큐비트에 각 비트(feature)를 할당

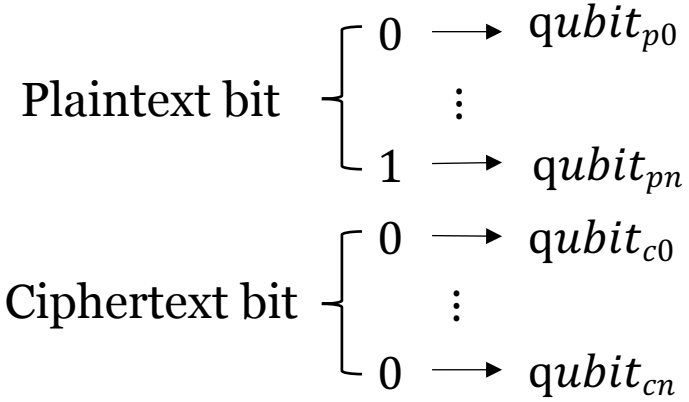
- **Quantum Support Vector Machine 사용**

  → 기존의 머신러닝 기법인 SVM의 커널 역할(비선형 함수)을 양자회로로 구현한 것

  → 즉, 해당 회로는 QSVM의 kernel (feature map)을 의미

- **이런 방식으로 다른 데이터들도 학습 가능**

Plaintext bit	Ciphertext bit			Key
0	1	0	0	3
⋮	⋮	⋮	⋮	⋮
1	1	0	1	2
Data				Label



*클라우드 환경 문제로 인해 2-bit, 3-bit 평문 및 키에 대해서만 수행했음

QSVM

- **Support Vector Machine (SVM)**
  - 초평면을 통해 **데이터 포인트 간의 최적 경계를 찾는** 지도 머신러닝 알고리즘
  - **분류 및 회귀에 사용**
  - **초평면** : n차원의 공간을 나누기 위한 n-1차원
  - **n차원 공간을 나누기 위해 kernel** 사용

    → kernel은 다양한 초평면을 잘 배치하여 공간을 잘 나눌 수 있도록 함

    → kernel 함수는 **데이터 포인트 간의 경계를 최대화하여 효율적으로 분리**하도록 함
  - 이러한 초평면을 찾기 위해서는 데이터에 **비선형 함수를 적용**해야 함

    → **feature map** 이라고 하며, 다항식, 시그모이드, 가우스 함수 등이 존재

- **QSVM은 SVM을 양자 회로로 구현하여 양자컴퓨터 상에서 동작**

  → 고차원의 데이터 작업에 유리하기 때문에 SVM이 처리하기 어려운 커널 최적화의 이점이 있음

  또한, 일반적으로 기존 SVM보다 성능이 좋음

QSVM

- **QSVM의 비선형 함수 φ**

$$\phi_S : x \mapsto \begin{cases} x_i & S = \{i\} \\ (\pi - x_i)(\pi - x_j) & S = \{i, j\} \end{cases}$$

- **Qiskit의 QSVM은 featuremap을 지원**

  Zfeaturemap (얽힘 없음), ZZfeaturemap(얽힘 존재), Paulifeaturemap (사용자 지정 게이트 사용 가능)

- 작성했던 논문에서는 Zzfeaturemap 사용

- **PauliFeatureMap**을 통해 사용자 지정 게이트 추가할 수 있음

  → 다른 게이트 사용 및 배치 가능

- **Optimizer**는 앞서 말한 3개 중 SPSA 사용

Parameterized Quantum Circuit

- **3비트 평문 및 암호문에 대한 파라미터화 된 양자 회로**

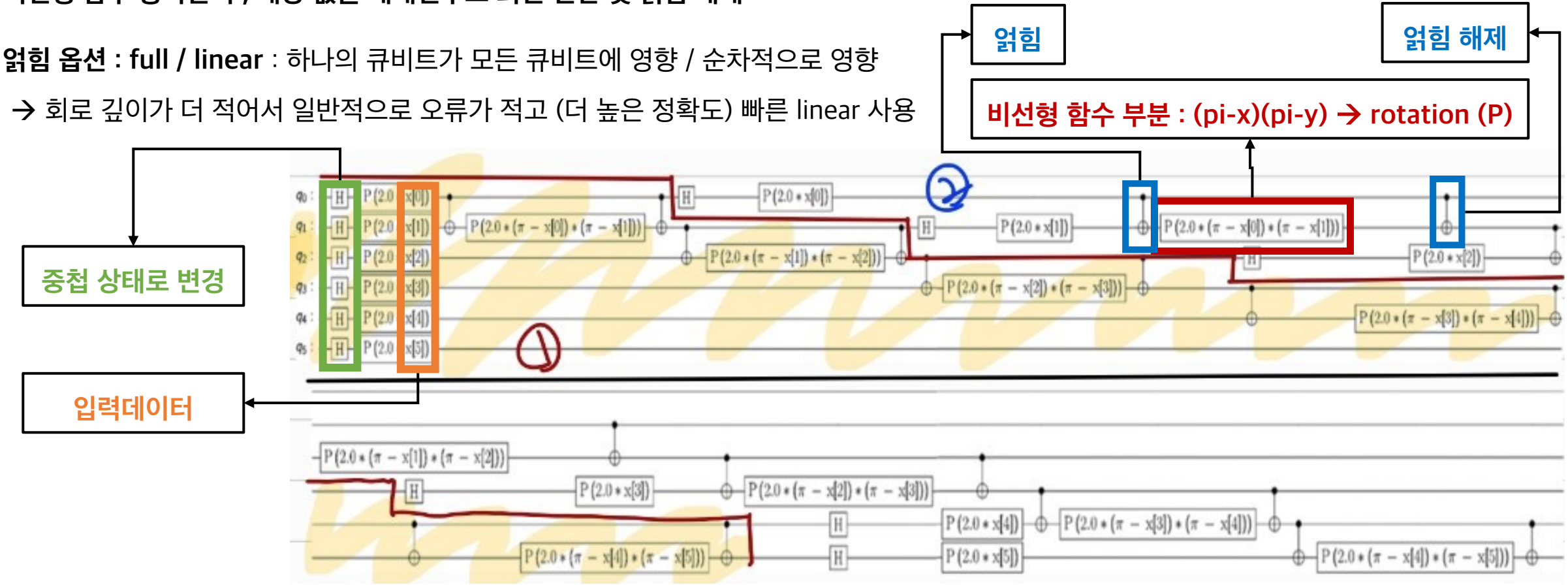
  - repetition = 2 로 설정하여 다음과 같이 2번 수행 → 입력이 총 6큐비트에 2번 반복하여 0.84 정확도 달성

  → 더 긴 평문 및 암호문 입력 시, repetition을 더 높게 설정 해야할 것 (큐비트가 많아지면 측정 확률이 떨어지니까)

  비선형 함수 통과한 후, 해당 값을 매개변수로 회전 연산 및 얽힘 해제

- **얽힘 옵션 : full / linear** : 하나의 큐비트가 모든 큐비트에 영향 / 순차적으로 영향

  → 회로 깊이가 더 적어서 일반적으로 오류가 적고 (더 높은 정확도) 빠른 linear 사용



실험 결과

입력 데이터 길이가 길어질수록 성능 저하

→ 회로 반복을 좀 더 해야할 듯

→ 수행 시간이 길어져서 시뮬레이터로 클라우드 상에서는 불가능 할 것 같음

Shots	2-bit dataset	3-bit dataset
1	0.66	0.6
5	1.0	0.7
100	-	0.81
150	-	0.84

Execution time	Accuracy
780	0.93

2-bit 데이터셋만 가능 (토큰이 없었음)

Q & A