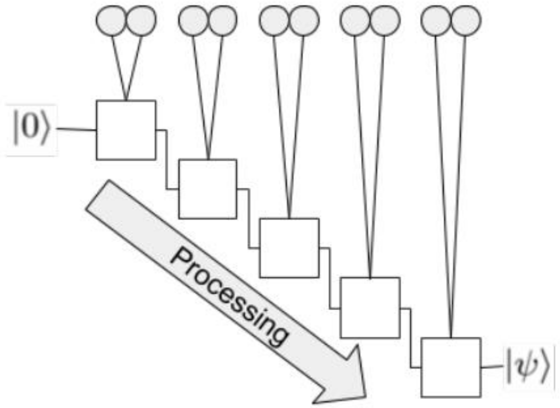
Quantum ensemble learning for classification with various quantum classifiers

SnuQuant

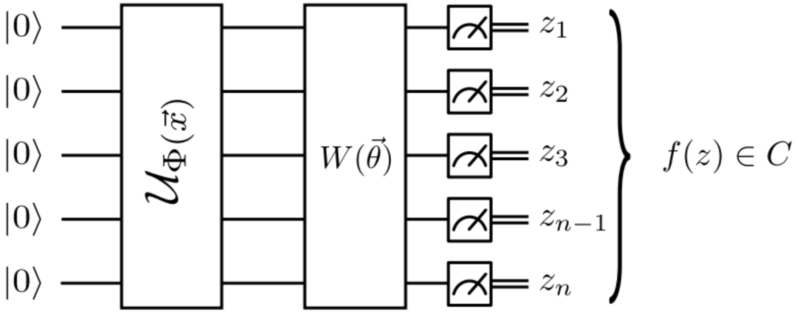
Jung, Jaewon

Quantum Classification Method

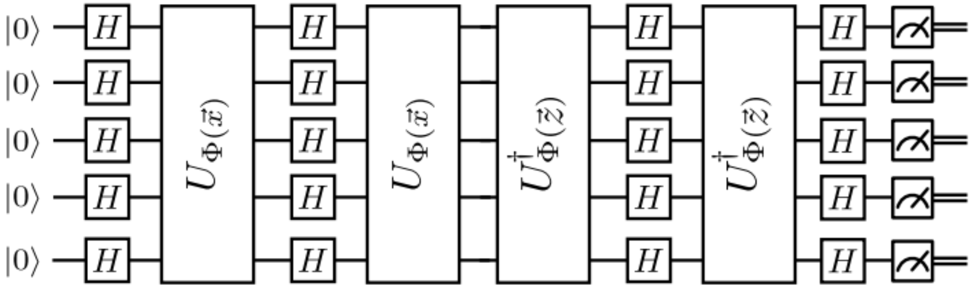
- **Supervised learning**
- - Quantum Support vector machine, Variational Quantum classifier, Data re-uploading method, QNN, QGANs….



Data re-uploading method



Variational Quantum Classifier

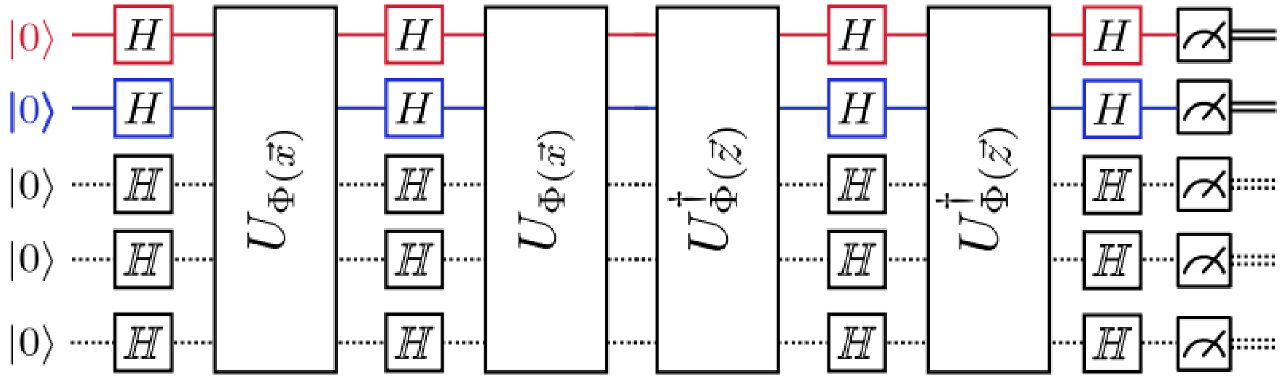


Quantum Support vector machine

Quantum support vector machine

1. Obtain Quantum kernel from direct inner product by quantum circuit using feature map

$$|\langle \Phi(\vec{x}) | \Phi(\vec{z}) \rangle|^2 \quad = \quad |\langle 0^n | \mathcal{U}^\dagger_{\Phi(\vec{x})} \mathcal{U}_{\Phi(\vec{z})} | 0^n \rangle|^2$$



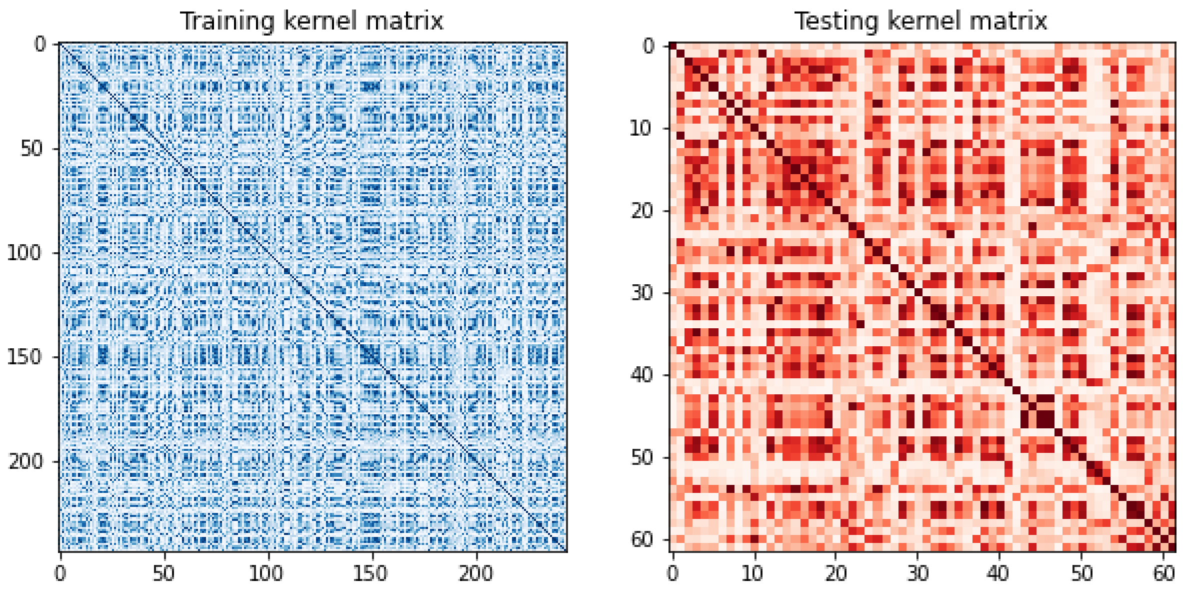
2. minimize $$L_D(\alpha) = \sum_{i=1}^{t} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{t} y_i y_j \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j),$$

$\sum_{i=1}^{t} \alpha_i y_i = 0$ and $\alpha_i \geq 0$ for each i.

Quantum support vector machine

3. From optimized $\vec{\alpha}^*$ get classification result

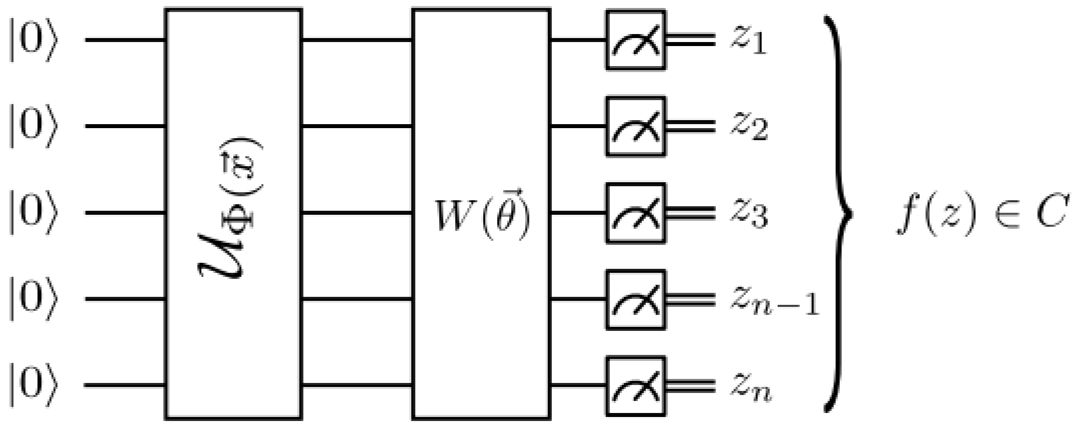
$$\tilde{m}(\vec{s}) = \text{sign}\left(\sum_{i=1}^{t} y_i \alpha_i^* K(\vec{x}_i, \vec{s}) + b\right).$$



Kernel from QSVM - statevector result
Training (left)
Test (right)

Variational quantum classifier

- 1. Encode classical input by feature map circuit
- 2. Apply variational circuit W($\vec{\theta}$) and get measurement



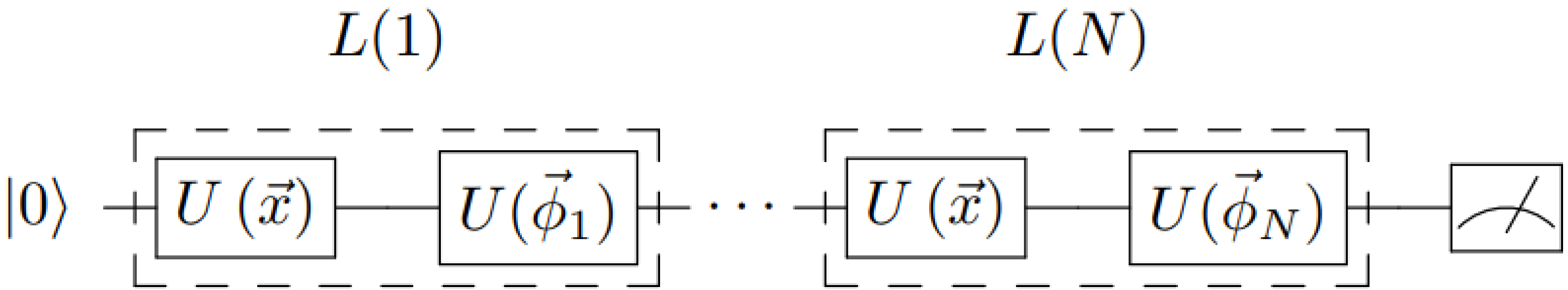
- 3. Optimize $\vec{\theta}$ from measurement result
- 4. from final optimized $\vec{\theta}$, get classification result

Data re-uploading method

- Series of data encoding and single qubit operations based on universality of arbitrary single qubit rotation

$$R(\phi,\theta,\omega) = RZ(\omega)RY(\theta)RZ(\phi) = \begin{bmatrix} e^{-i(\phi+\omega)/2}\cos(\theta/2) & -e^{i(\phi-\omega)/2}\sin(\theta/2) \\ e^{-i(\phi-\omega)/2}\sin(\theta/2) & e^{i(\phi+\omega)/2}\cos(\theta/2) \end{bmatrix}.$$

Matrix form of arbitrary gate



Layers of encoding and single qubit rotation

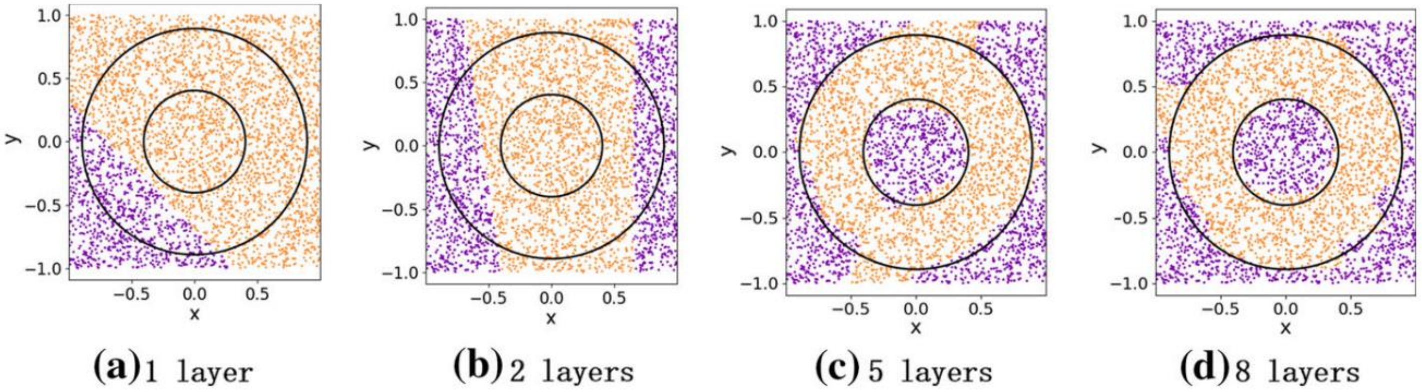
$$\chi_f^2(\vec{\theta},\vec{w}) = \sum_{\mu=1}^{M}\left(1 - |\langle\tilde{\psi}_s|\psi(\vec{\theta},\vec{w},\vec{x_\mu})\rangle|^2\right)$$

Fidelity cost function

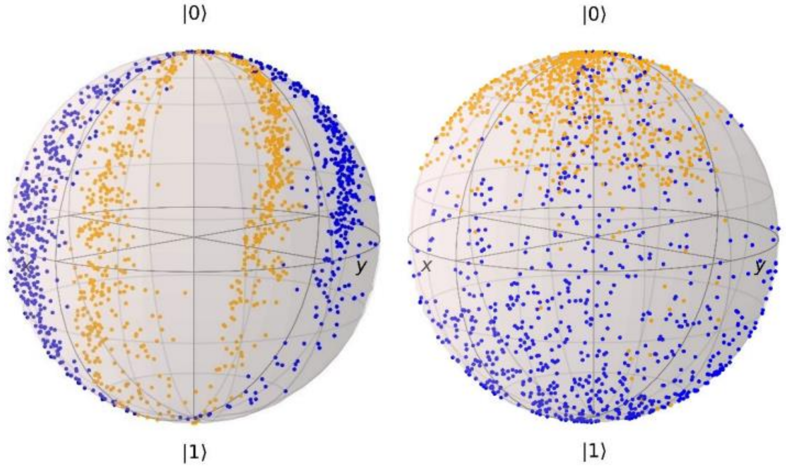
$$\chi_{wf}^2(\vec{\alpha},\vec{\theta},\vec{w}) = \frac{1}{2}\sum_{\mu=1}^{M}\left(\sum_{c=1}^{\mathcal{C}}\left(\alpha_c F_c(\vec{\theta},\vec{w},\vec{x}_\mu) - Y_c(\vec{x}_\mu)\right)^2\right)$$

Weighted fidelity cost function

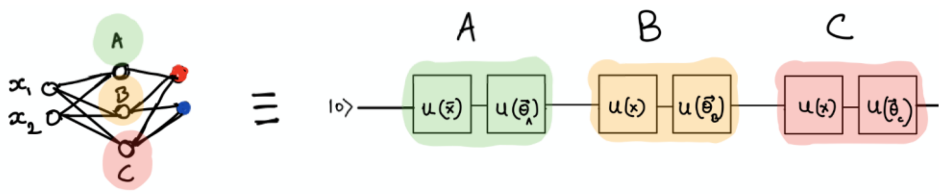
Data re-uploading method



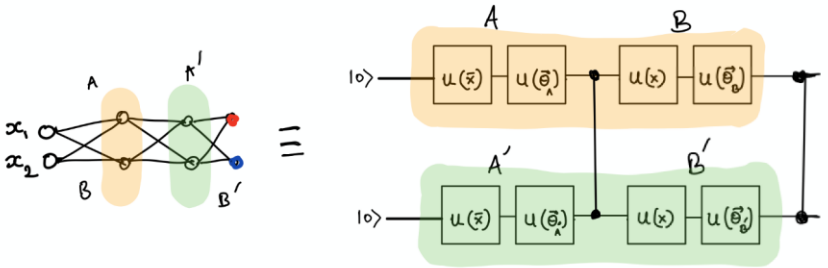
Classification of synthetic data



Bloch sphere representation of data



Only need one qubit regardless of input feature #
-> Decreased # of qubits and gates
(Of course, we can utilize more qubits if needed)



Data re-uploading method

- Compact data encoding – each layer has only one U gate

$$\phi_l = \langle \vec{w}_l, \vec{x} \rangle + b_l = \sum_{i=1}^{d} w_{l,i} \cdot x_i + b_l. \qquad L(l) = U(\phi_l, \phi_l, \phi_l).$$

- $L(l) = RZ(\varphi_l)RY(\varphi_l)RZ(\varphi_l)$
- Superconducting qubit, ion trap qubit support **Virtual-Z gate**
- We only need single physical gate $RY(\varphi_l)$ in each layer for implementation on real hardware.
- Saving a lot of quantum resources and time !

Data re-uploading method

- Compact data encoding
- Reducing circuit depth from 3 * # of layers to 2*# of layers+1
- $L(l) = RZ(\varphi_l)RY(\varphi_l)RZ(\varphi_l)$

For example, l=2

- $RZ(\varphi_2)RY(\varphi_2)RZ(\varphi_2)\ RZ(\varphi_1)RY(\varphi_1)RZ(\varphi_1)$
- => $RZ(\varphi_2)RY(\varphi_2)RZ(\varphi_2 + \varphi_1)RY(\varphi_1)RZ(\varphi_1)$
- 3*2 -> 2*2 +1
- As # of layers grow, we can benefit from this reduced compact circuit !

Dataset

- In the dataset selection process, we had to compromise between compactness and difficulty of classification
- Haberman’s Survival Dataset was selected
- 306 data points
- Features
- 1. Age of patient at time of operation (numerical)
  2. Patient's year of operation (year - 1900, numerical)
  3. Number of positive axillary nodes detected (numerical)
- Class : Survival status
  1 = the patient survived 5 years or longer
  2 = the patient died within 5 year

Quantum Ensemble learning with Quantum classifiers

If we know quantum state from classifiers,
We can also get benefit in Decision procedure.
We can use QST for ensemble vote also.

Input data

Classifier 1

Classifier 2

⋮

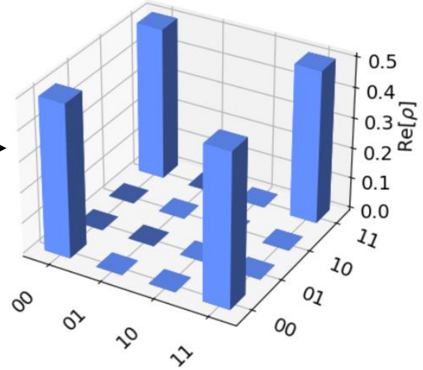
Classifier n

Decision procedure

Prediction

Voting, Quantum state tomography…

$$|\psi\rangle = \sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} c_{i_1 \dots i_n} |i_i\rangle \otimes \dots \otimes |i_n\rangle$$

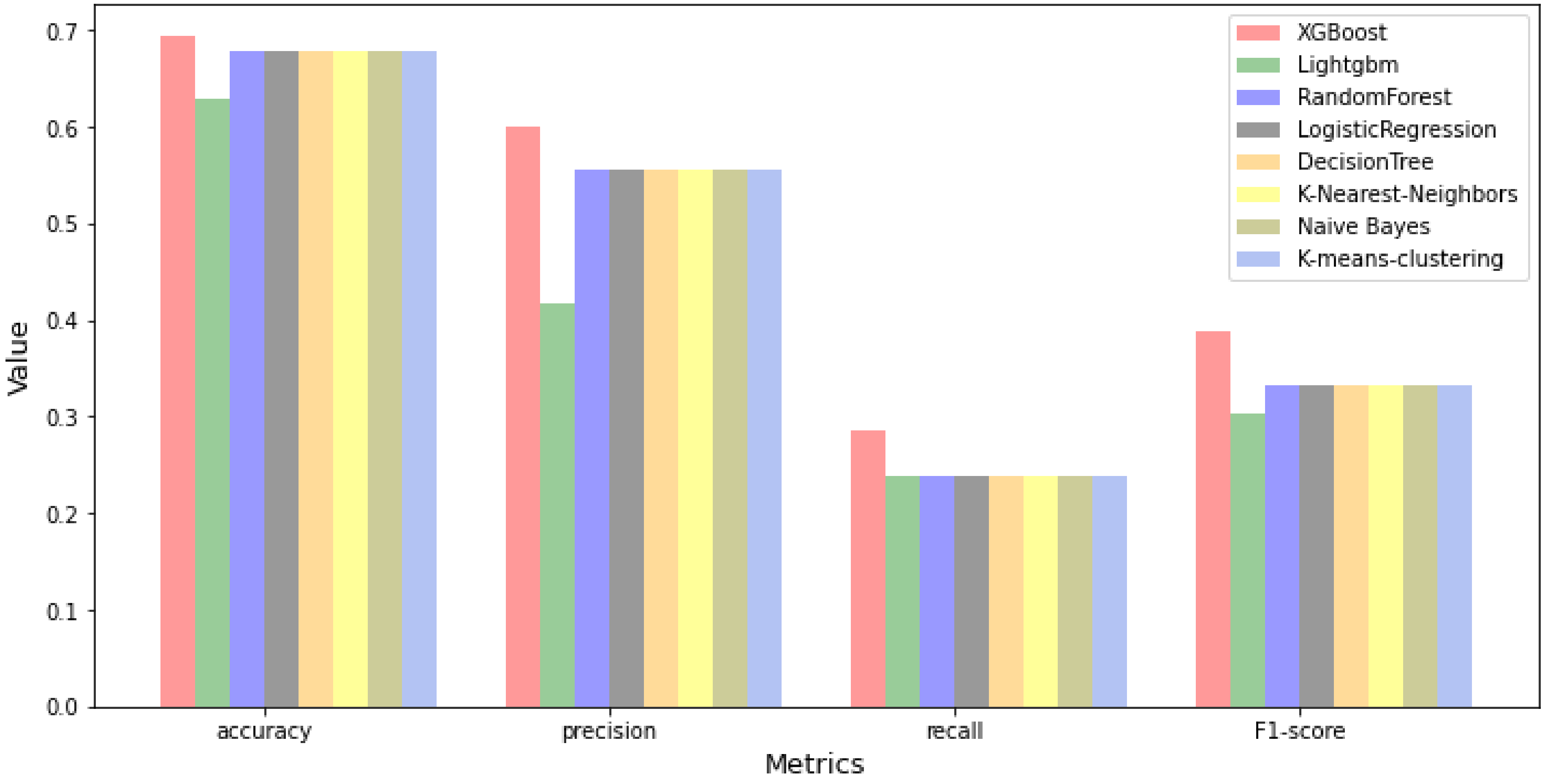


Data re-uploading method, QSVM, …..
Maybe some classical classifiers for hybrid classification

Quantum Ensemble learning with Quantum classifiers

- In decision procedure, we use
- 1. Simple averaging (y1+y2+….)
- 2. Weighted averaging (c1*y1+c2*y2+….)
- And do the majority voting from above -> obtain y_hat

Results from Classical algorithms



Results from quantum classifiers - Statevector

QSVM
accuracy: 0.7258
precision : 0.7000
recall : 0.3333
F1-Score : 0.4516

We used ZZFeaturemap(reps=2) for both QSVM, VQC
And used RealAmplitude(reps=3) ansatz for VQC

VQC
accuracy: 0.6774
precision : 0.5385
recall : 0.3333
F1-Score : 0.4118

Data re-uploading, 5 layers
With weighted fidelity(pennylane)
accuracy: 0.6935
precision : 0.5455
recall : 0.5714
F1-Score : 0.5581

Simple averaging
accuracy: 0.6774
precision : 0.5217
recall : 0.5714
F1-Score : 0.5455

Weighted averaging
accuracy: 0.7258
precision : 0.7000
recall : 0.3333
F1-Score : 0.4516

Results from quantum classifiers - Simulator

```
VQC-ionq_simulator
accuracy: 0.6774
precision : 0.5294
recall : 0.4286
F1-Score : 0.4737
```

```
VQC-ibm_Sherbrooke(aer)
accuracy: 0.6774
precision : 0.5294
recall : 0.4286
F1-Score : 0.4737
```

```
VQC-ibm_Sherbrooke(aer)
Applied DD sequence
accuracy: 0.6935
precision : 0.7500
recall : 0.1429
F1-Score : 0.2400
```

```
QSVM-ibm_Sherbrooke(aer)
accuracy: 0.6935
precision : 0.6000
recall : 0.2857
F1-Score : 0.3871
```

```
QSVM-ionq_simulator
accuracy: 0.7097
precision : 0.6667
recall : 0.2857
F1-Score : 0.4000
```

```
Simple averaging
accuracy: 0.7097
precision : 0.6364
recall : 0.3333
F1-Score : 0.4375
```

```
Weighted averaging
accuracy: 0.7258
precision : 0.7000
recall : 0.3333
F1-Score : 0.4516
```

Dynamical Decoupling's power check

These results are tested with ZZFeaturemap(reps=2), RealAmplitued(reps=5)
DD is applied on ansatz, we used long ansatz to see how DD is good for error mitigation

```
VQC-Statevector
accuracy: 0.7258
precision : 0.8333
recall : 0.2381
F1-Score : 0.3704
```

```
VQC-ibm_lima(aer) DD applied
accuracy: 0.7097
precision : 0.8000
recall : 0.1905
F1-Score : 0.3077
```

```
VQC-ibm_lima(aer) no DD
accuracy: 0.6452
precision : 0.0000
recall : 0.0000
F1-Score : 0.0000
```

For long ansatz, DD does matter!

Discussion

- From results, ensemble of quantum classifiers outperformed classical algorithms. (Still, we have to consider that we didn’t hyper tune classical algorithms)
- Nevertheless, it is shown that ensemble of quantum classifiers does have better classification result than single quantum classifiers.

Discussion

- Also, in simulator results, weighted averaging method improved the result from simple averaging.
- Surprising point is that even though each classifiers from simulator has worse performance than ones from statevector because of the presence of noise, when we did ensemble learning, classification power was ***same***. Therefore, we can think that ensemble learning not only improves the classification result, it is also robust to noise at certain level.

Discussion

- In Dynamical decoupling test, we can see that if the circuit is long enough to suffer noise, DD did mitigate errors.

Future works

- In this project, we didn’t select feature map, ansatz carefully, but in fact, picking feature map and ansatz is very important problem in QML.
- We have to pick and modify those circuits more elaborately.
- Also, in data re-uploading method, we can implement pulse other than gate because all we need in hardware implementation is RY gate. Utilising pulse, we can reduce circuit duration a lot shorter. Therefore, we can use more layers on noisy hardware.

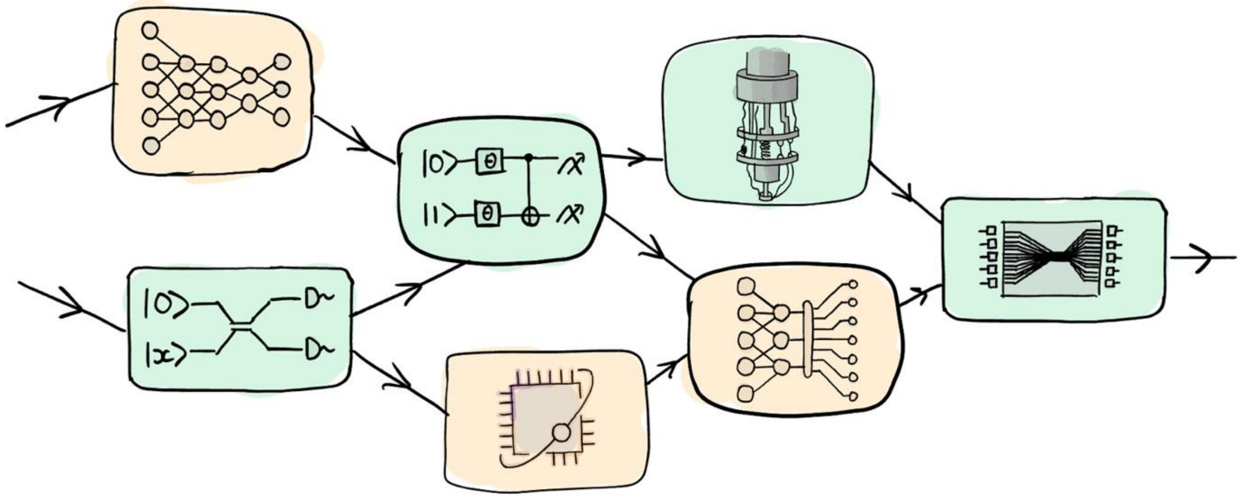
Future works

- For checking hardware capabilities, we couldn't see big difference between ionq_simulator and Ibm machine because of lack of testing.
- In later works, we can test them with using more controlled gate and more long circuits to check what kind of advantages are best suit for quantum classifying algorithms.
- All-to-all connectivity or good gate fidelity
-

References

1. Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." Nature 567.7747 (2019): 209-212.
2. Schuld, Maria, and Francesco Petruccione. "Quantum ensembles of quantum classifiers." Scientific reports 8.1 (2018): 2772.
3. Pérez-Salinas, Adrián, et al. "Data re-uploading for a universal quantum classifier." Quantum 4 (2020): 226.
4. McKay, David C., et al. "Efficient Z gates for quantum computing." Physical Review A 96.2 (2017): 022330.
5. Dutta, Tarun, et al. "Single-qubit universal classifier implemented on an ion-trap quantum device." Physical Review A 106.1 (2022): 012411.
6. Fan, Liangliang, and Haozhen Situ. "Compact data encoding for data re-uploading quantum classifier." Quantum Information Processing 21.3 (2022): 87.
7. Qin, Ruiyang, et al. "Improving Quantum Classifier Performance in NISQ Computers by Voting Strategy from Ensemble Learning." arXiv preprint arXiv:2210.01656 (2022)

Thanks



PENNYLANE

Qiskit

IONQ