テンソルネットワーク構造を用いた量子回路学習

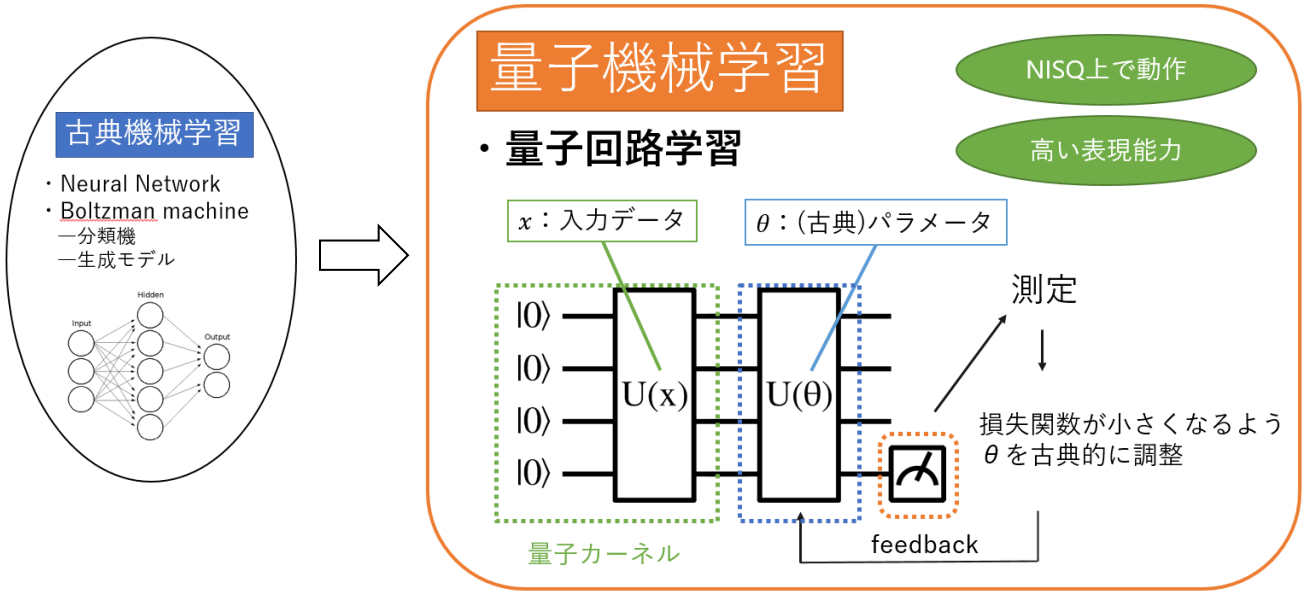
—QTensorNet：高速・簡単な量子回路学習ライブラリ—

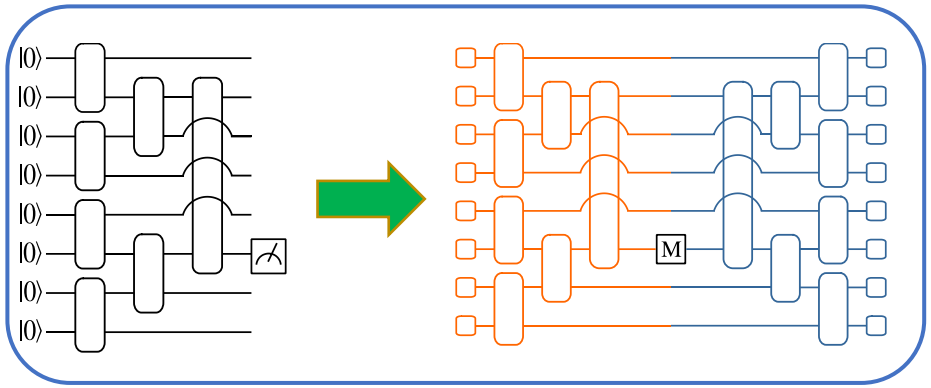
真鍋 秀隆（京都大学工学部情報学科）

Pythonライブラリの**QTensorNet**を用いて、量子計算・量子機械学習を手軽に体験することができます。

量子回路学習は量子コンピュータを用いた教師あり学習アルゴリズムで、古典コンピュータでは学習できないデータセットをも学習できる可能性があるとして近年注目されています。

さらに、**テンソルネットワーク**という枠組みを用いることにより、大規模量子回路のシミュレーションを高速に実現しています。





テンソルネットワークを用いた測定値の期待値計算

QTensorNetを用いた手書き文字認識実行例

①モジュールのインポート

```
import jax.numpy as np
import qtensornetwork.components
import qtensornetwork.circuit
import qtensornetwork.ansatz
import qtensornetwork.util
import qtensornetwork.optimizer
from qtensornetwork.gate import *
```

①各種モジュールのインポート

QTensorNetは，バックエンドとしてGoogleの**Jax**を使用.
通常のnumpyと同様に扱える.

②量子カーネルの設定

```
qnum = 64
circuit = qtensornetwork.circuit.Circuit(qnum)

xtrain, ytrain, xtest, ytest = generate_binary_mnist(0, 1, 1000, 200, 8, 8)

qxtrain = qtensornetwork.util.dtoq_miles(xtrain)
qxtest = qtensornetwork.util.dtoq_miles(xtest)

for i in range(qnum):
    circuit.set_init_state(qtensornetwork.components.State([i], None, train_idx=i))
```

②量子回路を準備

古典データを量子状態にエンコードする
いくつかの標準的な関数も使用可能.

③データを準備

④量子データと量子ビットを対応付ける

③量子ゲートの作成

```
def complex_gate():
    Rz0 = RZ(0,0)
    Ry1 = RY(0,0)
    Rz2 = RZ(0,0)
    Rz3 = RZ(1,0)
    Ry4 = RY(1,0)
    Ry5 = RZ(1,0)
    CNOT6 = CNOT([0,1])
    U_gate = combine_gates([Rz0, Ry1, Rz2, Rz3, Ry4, Ry5, CNOT6])
    return U_gate

cgate = complex_gate()
layer = qtensornetwork.ansatz.TTN([i for i in range(qnum)], gate_input_num=2,
                                  gate_output_num=1, gate_func=cgate.func, gate_params_num=6)
circuit.append_layer(layer)
```

標準的に用いる量子ゲートも使用可能.

既存のゲートを組み合わせて
新しいゲートを簡単に構成可能.

⑤テンソルネットワーク構造を用いて
回路を自動的に構成する.

④測定を追加・学習開始

```
m_tensor = np.array([[1, 0], [0, 0]])
measurement1 = qtensornetwork.components.Measurement(None, m_tensor)
circuit.add_measurement(measurement1)

circuit.show_circuit_structure()

optimizer = qtensornetwork.optimizer.Adam(lr=0.01)

circuit.classify(qxtrain, None, ytrain, qxtest, None, ytest,
                 optimizer=optimizer, epoch=50, batch_size=20)
```

⑥測定を準備

回路のおおまかな構造を描画する関数.

⑦optimizerを準備

⑧データとオプションを指定して学習開始

成果物はオープンソースで公開中

https://github.com/wotto27oct/QTensorNet