

フェーズ操作による量子機械学習

～汎用量子画像処理の新手法の確立に向けて～

著者：灘高等学校 3 年 林慶一郎

1. 要旨、概要

古典的な畳み込みニューラルネットワークから着想を得て、 $C^{\otimes n}RZ$ ゲートを用いて画像の入力や畳み込みを行う量子回路を設計し、量子画像認識機械学習を行なった。本研究の特色として以下の 7 点が挙げられる。

- (1) 重ね合わせた量子状態の測定確率に依存する係数にフェーズとしてデータを落とし込んだ点
- (2) 任意の画像サイズに対して入力を自動で行うアルゴリズムを開発した点
- (3) 畳み込み時の当手法特有の障壁をパディング法のアレンジにより克服した点
- (4) 同時摂動法を用いて古典計算の回数を極力減らした点
- (5) パラメータの更新は古典計算で行うハイブリッド型のアルゴリズムである点
- (6) 観測を複数回挟むことにより非線形性を担保した点
- (7) 使用する量子ビット数や畳み込み層が増えた場合にも対応した手法である点

MNIST の画像セットにおいて画像が 0 か 1 かを識別するモデルを設計し、Qiskit を用いて機械学習を行なった。精度面の検証を行い、新たな量子画像処理手法の理論の大枠を完成させるに至った。モデルの洗練により精度向上が期待され、量子データのみならず古典データを汎用的に多次元情報の処理を行う手法としての可能性を示す。

2. 問題提起、研究目的

本研究のきっかけは、著者の医用画像処理への興味から説明できる。脳の病に悩まされる著者は将来の診断技術として、人体を統合的に解析し総合診断を下すシステムを夢想した。体内の様子を CT や MRI を用いて画像情報として取得できるが、そこからは医師の判断に委ねられているのが現状である。医師の人為的な判断に、情報学的診断を加味することが臨床医学における最新動向であることは、第 194 回医用画像情報学会の東北大学医工学研究科の本間経康教授の講演でも示唆された。CT や MRI 等の画像情報ないし画像へと加工される前の情報から医師による診断精度を上回り総合診断を下すシステムを構想したときに、量子計算を用いたアルゴリズムに可能性を見出した。現状では一括では扱えない量の情報、過剰な処理プロセスを用いた情報処理に基づいた自動診断を実現するには、古典計算の枠から出て量子超越性に裏付けられた新たな情報処理手法の確立が必須であると考ええる。

本研究では、近年注目されている量子コンピュータの応用アルゴリズムの一つとして下記論文で提案されている量子機械学習、特に量子ニューロンの考え方に基づき、この量子ニューロンに対する入出力部分の具体化に関する未検討課題へのアプローチに始まり、量子アルゴリズムによる優位性を発揮できる学習モデルの実現、ハイブリッド型機械学習による重みの最適化に取り組む。

先行研究において量子機械学習の原理が提案されているが、実用例としてこれを画像認識に応用しようとした場合に、入力画像をどのように量子ニューロンへの入力とするのか、また、量子的なプロセスの結果からどのように「画像を認識できた」と判断するのか、といった「具体的な問題に適用しようとした場合に顕在化する課題」についてはこれまで十分に検討されているとはいえない。このような現実的な問題に向き合うことによって、新たに従来の手法との類似性や差異を顕在化させることが出来ると考えられる。また、画像判別のプロセスにおいて線形的手法による欠点に気がつけながら、複数の重み層を持つ構造を設計し、より実践的な画像認識機械学習を試みた。本研究ではノイズの影響を一旦無視し、量子プログラミングのためのオープンソースである Qiskit を用いてこの課題に取り組み、実用化の基盤を構築する。なお、実験に際して、NICT の提供する汎用開発環境を使用した。

3. 研究方法

Jupyter notebook 上に Python でコードを作成しオープンソースの量子回路生成・量子計算のフレームワークである Qiskit を用いて研究を行なった。「要旨、概要」に記載した本研究の特色とそれに付随する手法の研究方法ごとに説明をし、適宜付け加えをするものとする。

(1) 重ね合わせた量子状態の測定確率に依存する係数にフェーズとしてデータを落とし込んだ点

量子ビットの重ね合わせはアダマールゲートによって以下のように表される。

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (1.1)$$

これを Z 基底で観測すると $|0\rangle$ と $|1\rangle$ が観測される確立はそれぞれ $1/2$ である。一般に、規格化された α, β において $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ が与えられた時、 $|0\rangle$ の観測確率は $|\alpha|^2$ で $|1\rangle$ の観測確率は $|\beta|^2$ である。このことから着想を得ると、(1.1)式は初期値が $|1\rangle$ である場合を鑑みて一般化すると、

$$H|x_k\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \exp\left(\frac{2\pi i}{2}x_k\right) \end{pmatrix} \quad (1.2)$$

と表現可能である。これに $RZ(\theta)$ ゲートを導入すると、

$$Rz(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \quad (1.3)$$

で表されるため、(1.2)式の状態を $|\psi\rangle$ とすると、

$$R_z(\theta) |\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} \exp\left(-i\frac{\theta}{2}\right) \\ \exp\left(i\frac{2\pi x_k + \theta}{2}\right) \end{pmatrix} \quad (1.4)$$

で表される。画像情報の入力や処理は、 $C^{\otimes n} R_z(C^{\otimes n})$ (は n 個のコントロールビットを持つという意) を用いてこの θ の値を操作することで行う。詳しくは (2) や (3) を参照されたい。

(1.4) 式をそのまま観測しても $|0\rangle$ と $|1\rangle$ の観測確率はそれぞれ $1/2$ であることに変わりはない。そこで、量子フーリエ変換(QFT)を導入する。量子フーリエ変換は、波動関数の振幅に対する離散フーリエ変換の量子アルゴリズムによるものである。離散フーリエ変換においてベクトル $(x_0, x_1, \dots, x_{N-1})$ のベクトル $(y_0, y_1, \dots, y_{N-1})$ へのマッピングは

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j w_N^{jk} \quad (1.5)$$

に定義される。ここで、 $w_N^{jk} = e^{2\pi i \frac{jk}{N}}$ である。一方で量子フーリエ変換は

(1.5) 式と同様の x と y の対応関係に則って量子状態 $\sum_{i=0}^{N-1} x_i |i\rangle$ を

量子状態 $\sum_{i=0}^{N-1} y_i |i\rangle$ にマッピングする。

$$|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} w_N^{xy} |y\rangle \quad (1.6)$$

ユニタリ行列としては、

$$U_{QFT} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} w_N^{xy} |y\rangle \langle x| \quad (1.7)$$

と表現される。

量子フーリエ変換は計算基底(Z基底)とフーリエ基底(eの指数部分に依存す

る)の二つの基底を変換する操作である。Z 基底の状態 $|0\rangle$ と $|1\rangle$ を x 基底の状態 $|+\rangle$ と $|-\rangle$ に変換する。同様に、計算基底の全ての量子ビットは対応するフーリエ基底の状態を持っている。図 1 のブロッホ球において、観測確率とは z 軸へ写像した際の点の座標の 2 乗に依存する量のことである。その意味で z 軸周りの回転操作で xy 平面上を動く入力や畳み込み操作をしたところで図 1 においては一向に $|0\rangle$ と $|1\rangle$ の観測確立が半々であることが理解できる。これに対し、量子フーリエ変換は基底を入れ替えることで xy 平面から脱し、固有の観測値を与えるものである。(1.2)式の H ゲートは 1 量子ビットの量子フーリエ変換であるというのは、直感的に分かりやすい例であると言える。本研究においては、離散的に古典データを入力し画像処理をする過程をフェーズ規定で行い、観測に際して量子フーリエ変換でフェーズを取り出すという流れである。それを以下に示す。

(2) 任意の画像サイズに対して入力を自動で行うアルゴリズムを開発した点

古典的な機械学習におけるニューラルネットワークでは、入力情報をノードにマッピングし、重みを掛け合わせ、それが閾値を上回れば発火するというプロセスを何度も行い、最終的な出力から元の入力情報を分類したり、値を予測したりする(図 2)。本研究において入力する情報をどのように量子状態にマッピングするか、またどのように重みを掛け合わせるかについての着想は、先行研究から得た[1].入力情報がマッピングされたベクトル \vec{i} をとし、それに掛け合わせる重みのベクトル \vec{w} をとす。

$$\vec{i} = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{m-1} \end{pmatrix}, \vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix} \quad (2.1)$$

このとき $i_j, w_j \in \{e^{i\theta} | \theta \in \mathbb{R}\}$ で、量子状態 $|\psi_i\rangle, |\psi_w\rangle$ を、

$$|\psi_i\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j |j\rangle, |\psi_w\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j |j\rangle \quad (2.2)$$

で表す。このとき、 $|j\rangle \in \{|00\dots00\rangle, |00\dots01\rangle, \dots, |11\dots11\rangle\}$ であり、この 0 と

1 のみの数列は 10 進数である j の 2 進数表記であると解釈する。量子ビットはそれぞれ確率的に $|0\rangle$ と $|1\rangle$ が観測される状態の重ね合わせ状態であり、 $|\psi_i\rangle, |\psi_w\rangle$ はこれらの複数の量子ビットの状態を表す。この考えのもと、簡易的な量子回路の構造を考える (図 3)。ここで U_i は初期状態 $|0\rangle^{\otimes n}$ から $|\psi_i\rangle$ の量子状態を作るユニタリゲートである。また U_w は $|\psi_w\rangle$ で表される重みを掛け合わせるユニタリゲートである。最終的に発火を $C^{\otimes N}X$ ゲートによる Ancilla qubit の反転に委ねるものとする。

$$U_i|0\rangle^{\otimes N} = |\psi_i\rangle, U_w|\psi_w\rangle = |1\rangle^{\otimes N} = |m-1\rangle \quad (2.3)$$

このとき、 $m=2^N$ である。 U_w が $|\psi_w\rangle$ という重みをかけるとき、 U_w が適用される直前の量子状態が $|\psi_w\rangle$ ならば、 U_w 適用後の N 個の量子ビット全てが $|1\rangle$ になり $C^{\otimes N}X$ ゲートは Ancilla qubit を反転し、測定値は 1 になる。入力情報に対応した $|\psi_i\rangle$ に U_w が適用されるとき、

$$|\phi_{i,w}\rangle \equiv U_w|\psi_i\rangle = \sum_{j=0}^{m-1} c_j |j\rangle \quad (2.4)$$

(2.3) 式と (2.4) 式の定義を用いると、二つの量子状態の内積は、

$$\langle \psi_w | \psi_i \rangle = \langle \psi_w | U_w^* U_i | \psi_i \rangle = \langle m-1 | \phi_{i,w} \rangle = c_{m-1} \quad (2.5)$$

である。 $|c_j|^2$ は $C^{\otimes N}X$ ゲートが適用される直前に N 個の量子ビットの量子状態が $|j\rangle$ である確率を示す。そのため $|c_{m-1}|^2$ は量子状態が $|11\dots 11\rangle$ である確率、つまり Ancilla qubit が反転する確率を示す。

まず初めに入力する画像の各ピクセルのグレースケールを規格化する。ここで図 4 のようにフェーズとグレースケールを対応させるため、

グレースケール g ($0 \leq g < 256$) に対して、フェーズは $\theta = \frac{g}{256}\pi$ で表す。よっ

て $i_j \in \{e^{i\theta} | \theta \in \mathbb{R}\}$ と表される。 $n \times n$ のピクセル数の画像があったとして、この画像を格納するための量子ビット数は $\lceil 2\log_2 n \rceil + 1$ 個である (ここで $\lceil \cdot \rceil$ はガウス記号)。この $n \times n$ の θ に規格化したグレースケールを格納し、それを下に画像の入力を行う。パディングについては (3) を参照されたい。

ここで例として図 5 のような 2×2 の画像について検討する。この画像は 2 量子ビットに情報を収めることができるが、その手法について説明する。

$C_1^{\otimes n} R_z(\theta)$ ゲートを導入すると、 $|j\rangle$ という量子状態に対して、

$$C_{\vec{l}}^{\otimes n} R_{z_k}(\theta) |j\rangle = \begin{cases} e^{i\left(\frac{\pi}{2} - \frac{\theta}{2}\right)} |j\rangle & \text{if } \bigcap j_i \geq l_i \text{ and } j_k = 0 \\ e^{i\left(\frac{\pi}{2} + \frac{\theta}{2}\right)} |j\rangle & \text{if } \bigcap j_i \geq l_i \text{ and } j_k = 1 \\ |j\rangle & \text{otherwise} \end{cases} \quad (2.6)$$

と表される。ここで、 \vec{l} の m 番目の要素は m 番目の量子ビットがコントロールビットならば 1 を、そうでなければ 0 を格納し、 k は k 番目の量子ビットがターゲットビットであることを示している。コントロールビットが全て 1 の時にターゲットビットに $R_z(\theta)$ が適用されるが、 k 番目の量子ビットが 0 か 1 によって途中の符号が変わる((1.3)式参照)。この k 番目の量子ビットが $|0\rangle, |1\rangle$ である確率は、この量子ビットが xy 平面上にいる限りそれぞれ $1/2$ ずつである。ここで、図 6 のような二つの CRZ ゲートをそれぞれに H ゲートを適用させた後の 2 量子ビットに用いると、

$$|\psi\rangle = \frac{1}{2} \left(|00\rangle + e^{i\left(\frac{\pi}{2} - \frac{\varphi}{2}\right)} |01\rangle + e^{i\left(\frac{\pi}{2} - \frac{\theta}{2}\right)} |10\rangle + e^{i\left(\pi + \frac{\varphi}{2} + \frac{\theta}{2}\right)} |11\rangle \right) \quad (2.7)$$

が得られる。この e の係数部分を入力に対応したフェーズ \vec{l} になるように、行列から一次連立方程式を設計し、それを解くことで R_z ゲートのパラメータを確定させる。このとき(2.7)式の例では 3 つの入力値に対してパラメータが二つのため連立方程式が解けないが、

$$\sum_{k=2}^n (nCk) \times k > 2^n \quad (n > 2) \quad (2.8)$$

より、3 量子ビット以上では R_z ゲートの種類の方が入力値の数より大きくなるため、この手法を使うことができる。これに応じて U_i, U_w の中身の量子回路を自動で設計するアルゴリズムを完成させた。具体的なコードは割愛するが、別個に作成した二つについて、入力情報に対応した $|\psi_i\rangle$ に U_w が適用されると 1 を出力する(2.5)ことから、正しく回路を組めていることを確認できた。

(3) 畳み込み時の当手法特有の障壁をパディング法のアレンジにより克服した点

MNIST を用いて画像診断を行なった。MNIST の画像は 28×28 のサイズのため、画像の周囲に画素を付け足すことで(パディング)、 32×32 の画像に前処理をした。これにより、入力に必要な量子ビットは 10 個になる。

畳み込みとは画像にフィルターを当てて重なった部分の数値を掛け合わせてそれを足し合わせることを繰り返す、とよく説明される。本研究では画像に対し、重ね合わせの係数を図 7(6 量子ビットの場合)のように対応させ、量子回路は図 8 のような形式を提案する。図 7 について、例えば左上の 4 マスは上から 0,1,3,4 番目(0 から数え始める)の量子ビットは全て 0 で共通である。ここで、この四つの量子ビットに X ゲートと $C_{(110110)}^{\otimes 4} R_z(\theta)$ ゲートを適用すると、0,1,3,4 番目の量子ビットが初め 0 だったときにだけ新たに導入したターゲットビットに R_z ゲートが作用する。このとき、新たに導入した量子ビットの一部もコントロールビットとする。入力に使った 10 量子ビットを無視したときに毎回異なる $C^{\otimes n} R_z(\theta)$ を用いることで、畳み込み処理ごとに独立な情報として新たに導入した量子ビット群に情報をマッピングする。このときの追加のコントロールビットとターゲットビットは入力画像の位置関係をそのまま保持する形でマッピングする必要はない。仮に位置関係を保持する形でマッピングするとしても、左上から順次

$|00...00\rangle, |00...01\rangle, \dots, |11...11\rangle$ の順に係数のフェーズをコントロールできるのなら画像情報の入力時に連立方程式を設計して解くコードを書く必要はない(実は 1 番苦労したところ)。一つの量子状態を個別に操作することができないのが本研究で最も苦労した点と言っても過言ではない。位置関係は畳み込みのコントロールビットの相関がある程度の範囲で位置関係の情報を保持しており、これ以上に位置関係にこだわる必要は薄く、むしろヒルベルト空間上で距離の離れた情報に関連を見出すためでもある。どちらにせよ一般的な古典ニューラルネットワークや本手法においても最終的に全結合層は全ノードを全探索的に繋ぎ合わせるため位置関係の情報は失われる。

ここで畳み込みを実装する際、本研究で考えた量子アルゴリズム特有の障壁に直面した。図 9 で示すような格子を跨ぐようにはフィルターを当てられない。この例によれば、格子内の量子状態は半分の量子状態が一致しているが、格子を跨ぐと一致する数が変わる。強引に同じ手法でコントロールビットを設定すると、同じ条件に当たる 4 マスの組み合わせが他に見つかることが確認できる。これではフィルターを当てた部分の情報を独立に取得することができない。格子で隔てられた画素の関係は完全に無視しなければならないとも考えられたが、これを克服する小技として、画像を入力する前の処

理として行うパディングに着目した。元々 28×28 のMNIST画像の外周を2マスずつ黒いマスを付け足すことで 32×32 の画像に加工するのが常だが、上下左右に付け足すマスを変化させることで幾通りかの画像を作ることができる(図10)。これらの画像は一辺が m の正方形の画像であれば

$\left(2^{\lceil \log_2 m \rceil + 1} - m\right)^2$ 通りの自由度があるが、これをフィルターのマスの数だけ

用意すると図9の格子を跨ぐ画素同士の畳み込みを行うことができる。ここで、各加工パターンにおいて同じ m 番目の画像の情報をヒルベルト空間内での距離が近くなるように $C^{\otimes n} R_z(\theta)$ ゲートのコントロールビットとターゲットビットを選ぶようにした。

これらを踏まえて、画像の加工ステップを示すと、図11のようになる。Pooling層と表記しているのは図8における最後の量子フーリエ変換の操作のことで、データサイズを減らしつつフェーズ基底の情報を古典的な観測値として得るためにX基底へと変換している。Convolution層とPooling層を繰り返す既存のやり方に、非線形的操作である観測を挟みそれを量子状態に投入し直すというステップを付け加えた。これについては(6)も参照されたい。0と1だけからなる画像データを学習し、最終的にある画像が1である確率を全結合層を経て出力する。

(4) 同時摂動法を用いて古典計算の回数を極力減らした点

図11に示される本研究の画像加工ステップにおいて、機械学習の勾配降下法で更新するパラメータの数は1292こ($8 \times 8 \times 16 + 4 \times 4 \times 16 + 2 \times 2 \times 2 + 4$)である。あるトレーニングデータ i とパラメータ群 \vec{k} から得られる出力結果の期待値 $E(i, \vec{k})$ を

$$E(i, \vec{k}) = \frac{1}{\kappa} \sum_{m=0}^{\kappa-1} E_m'(i, \vec{k}) \quad (4.1)$$

で表す。ここで κ はshot数であり、本研究では $\kappa=1024$ に設定した。また m 個ずつの0と1のトレーニングデータに対し、損失関数 L を

$$L(\vec{k}) = \frac{1}{2m} \sum_{j=0}^{m-1} \left[\left\{ E(i_{zero,j}, \vec{k}) \right\}^2 + \left\{ 1 - E(i_{one,j}, \vec{k}) \right\}^2 \right] \quad (4.2)$$

と設計する。本研究では各トレーニングデータの数 は 100 ずつとした。この損失関数の値を最小化する方向に各パラメータを更新していく。t 回目の学習のパラメータ群 \vec{k} を \vec{k}_t と表記することになると、

$$\vec{k}_{t+1} = \vec{k}_t - \eta \Delta \vec{k}_t \quad (4.3)$$

$$\begin{aligned} \Delta k_{t,j} &= \frac{L(\vec{k}_t + c\vec{s}_t) - L(\vec{k}_t - c\vec{s}_t)}{2cs_{t,j}} \\ &= \frac{L(\vec{k}_t + c\vec{s}_t) - L(\vec{k}_t - c\vec{s}_t)}{2c} s_{t,j} \end{aligned} \quad (4.4)$$

ここでは、同時摂動法を用いてパラメータを一度更新する時に損失関数の計算は 2 回で済むようにしている。 $s_{t,j}$ は 1 または -1 で、ベルヌーイ分布に基づき各要素は独立にランダムに決定される。 $c(>0)$ は全ての要素に共通の摂動の大きさを表している。愚直にパラメータの更新を行うと、

$$L'(\vec{k}_t) = \left(\frac{\partial L(k_{t,0})}{\partial k_{t,0}}, \frac{\partial L(k_{t,1})}{\partial k_{t,1}}, \dots, \frac{\partial L(k_{t,1451})}{\partial k_{t,1451}} \right)^T \quad (4.5)$$

の計算をすることになり、同時摂動法を用いることで損失関数の計算回数を大きく削減している。

同時摂動法は単なる計算量削減のための”妥協”ではなく、期待値を取ると各パラメータについて偏微分した値を算出している。これについては Appendix を参照されたい。

(5) パラメータの更新は古典計算で行うハイブリッド型のアルゴリズムである点

(4) で示したように、量子回路におけるパラメータを、勾配降下法で損失関数の値を最小化する方向に更新している。その過程は計算量を減らした古典計算によるもので、古典と量子のハイブリッド型のアルゴリズムであるといえる。

(6) 観測を複数回挟むことにより非線形性を担保した点

図 11 の学習ステップにおいて、Convolution 層と Pooling 層の後に観測を行いその値を再度入力することで線形性を回避している。この理由は、量子ゲートはどれもユニタリ変換を施すため線形的な演算である。そのため量子ゲートを多用して層を深くしても、一つのユニタリゲートとして見なすことができ、学習精度が頭打ちになる可能性がある。わかりやすい例として、ルービックキューブは全面が揃った状態からたくさん混ぜることができるが、どれだけ混ぜても整った状態から 10 手でその状態を作ることができるのと同じである。

(7) 使用する量子ビット数や畳み込み層が増えた場合にも対応した手法である点

研究手法を考えるにあたって、MNIST だけではなく任意のサイズの画像、または画像以外のデータに対して行える汎用的な量子機械学習手法であることを心がけた。

4. 結果

上記の研究方法にて示した方法に基づき Python で実装したが、26qubit を扱う長大なコードになり、手元のコンピュータはもとより、NICT の汎用開発環境の 32 コアを用いて尚メモリが足りずカーネルがクラッシュした。そこで、図 12 に示す学習ステップで再度試みた。28×28 の MNIST 画像を 8×8 に圧縮し、パディングで 16×16 の画像を 1 枚の元画像あたり 4 種類作成し、10qubit にマッピングし、Convolution 層の段階で 18qubit を使い、最終的に 8qubit を観測して出力をする。それらを 4×4 の画像として整形し、さらにパディングで 8×8 の画像に整形する。この流れを 32 コアで並列処理をし、1 度の学習に 2 時間のペースで 100 回更新をした。損失関数の推移は図 13 に示す。順調に数値は収束していったが、学習精度は 53% だった。原因は、16×16 のデータのうち 3/4 がパディングに当てられた”空白”であることが考えられた。そこで、さらに図 14 に示す学習ステップで試みた。28×28 の MNIST 画像を 7×7 に圧縮し、パディングで 8×8 の画像を 1 枚の元画像あたり 4 種類作成し、8qubit にマッピングし、Convolution 層の段階で 14qubit を使い、最終的に 6 つの出力をする。それでも学習精度は 55% だった。”空白”部分が多いという問題に着目したのは正しいはずだが、2 周目に入る前の 4×4 のデータへの整形が適切ではないことが考えられる。そこで図 15 に示す学習ステップを試みた。2 分類問題のニューラルネットワークでは各層ごとに徐々に特徴量を顕著にするもので、1 周目の量子アルゴリズムによる操作が正

しければ、その後で古典アルゴリズムに引き渡すと分類ができるはずである。ここで、古典アルゴリズムで各テストケースにおいて 0 か 1 である確率を計算できるため、正答との誤差の 2 乗を集約し、それを損失関数とした。ここで、古典アルゴリズムとして PyTorch を採用した。

結果として、初めは 50% の精度だったものが、73% まで向上した。

5. 考察

当手法とその結果について論じる前に、量子ゲートの”深さ”の概念を導入する必要がある。ユニタリー変換を行う各量子ゲートは、2 量子ビットがいくつか組み合わさった形に分解することが可能である。分解した際に必要となる 2 量子ビットの個数を”深さ”と定義する。幾つか存在するオープンソースによって、バックグラウンドでの分解手法が異なるため、各量子ゲートの深さも異なる。今回用いた Qiskit では図 16 に示すように、ある量子ゲートに関わる量子ビット数を n とするとその量子ゲートの深さは $O(k^n)$ ($2 < n < 3$) のオーダーである。一方で、図 16 に示される Linear 法では $O(n^2)$ のオーダーで分解が可能である。

さて、当手法をはじめ多くの量子アルゴリズムの評価で一つの指標となるのは計算コストである。ここで、計算コストとは使用する量子ゲートの深さの和として考えられるため、上で論じた深さと使用する量子ゲートの数を減らす努力が必要である。当手法では、古典的手法におけるフィルターの重ね合わせ一度に対し一つの量子ゲートを用いる。元データ量が 2^n でフィルターのサイズが 2^m で表される時、この操作は 2^{n-m} 回行われる。同じ条件の古典的手法では m が n に対して十分に小さい時、 2^n 回に近似できる。またこの条件下で、一つの量子ゲートの深さは、古典的手法における掛け算の数と同じ 2^n である。よってこれらの積である計算コストは、最小ゲート操作にかかるコストを古典計算における掛け算にかかるコストの T 倍とすると、古典的手法に比べて当手法では、 $2^{-m} \times T$ 倍として考えることができる。

今回 MNIST を解くのに同時に使う最大量子ビット数は、図 11 の元画像の場合では 26 個である。このサイズの量子回路は決して構築が不可能なわけではないが、古典コンピュータによるシミュレートで実行するには当手法は重すぎる。試行錯誤の末に図 15 のモデルに辿り着き、量子ビットの数を 14 個に落とすことができた。28×28 の MNIST 画像を 7×7 に圧縮して機械学習にかけたわけだが、古典的な機械学習で 88% だったものが量子計算で 73% の精度で分類できたことは大きな成果であると言える。

量子機械学習による古典的データの処理の長所を大量の入力データを処理できること

にあるとするならば、今回の図 15 のモデルにおいて、第一層の量子機械学習の出力は入力データのサイズに対して指数関数的に減少するため、第二層で量子アルゴリズムを用いる必要はないとも考えられる。本研究で手法の有用性を検証するために一層だけ量子アルゴリズムを用いたモデルを考えたが、偶然にもこのモデルこそ当手法を最大限活用できるものかもしれない。

6. 結論(課題)

本研究では 1 層だけ量子アルゴリズムを用いた機械学習により、量子回路で用いるパラメータだけの更新により 50%だった精度を 73%に引き上げることができた。当手法はデータ量が増えた際や多次元データを扱う際に有効であり、古典的手法よりも明らかに計算量を軽減できることが期待される。量子ビットを増やした際の精度面の評価が今後の課題である。

7. 参考文献

- [1] F. Tacchino, C. Macchiavello, D. Gerace, D. Bajoni, “An artificial neuron implemented on an actual quantum processor,” npj Quantum Information 5, 1-8 (2019).
- [2] S. Mangini, F. Tacchino, D. Gerace, C. Macchiavello, D. Bajoni, “Quantum computing model of an artificial neuron with continuously valued input data,” Mach. Learn.: Sci. Technol. 1 045008 (2020).
- [3] F. Tacchino, C. Macchiavello, D. Gerace, D. Bajoni, “Variational learning for quantum artificial neural networks”
- [4] F. Tacchino, P. Kl. Barkoutsos, C. Macchiavello, D. Gerace, I. Tavernelli, D. Bajoni, 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), 130-136 (2020).
- [5] F Tacchino, P. Barkoutsos, C. Macchiavello, I. Tavernelli, D. Gerace, “Quantum implementation of an artificial feed-forward neural network,” Quantum Sci. Technol. 5 044010 (2020).
- [6] S. Mangini, F. Tacchino, D. Gerace, D. Bajoni, C. Macchiavello, “Quantum computing models for artificial neural networks” EPL (Europhysics Letters) 134, 10002 (2021).
- [7] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, “Parameterized quantum circuits as machine learning models,” Quantum Sci. Technol. 4 043001 (2019).

- [8] A. Chalumuri, R. Kune, B. S. Manoj, “Training an Artificial Neural Network Using Qubits as Artificial Neurons: A Quantum Computing Approach,” *Procedia Computer Science*, 171, 568-575 (2020).
- [9] Y. Li, R-G. Zhou, R. Xu, J. Luo, and W. Hu, “A quantum deep convolutional neural network for image recognition” *Quantum Sci. Technol.* 5 044003 (2020).
- [10] Da silva, A.j., &Park, D.K. (2022). Linear-Depth Quantum Circuits for Multi-Qubit Controlled Gates. *Phys. Rev. A* 106, 042602

8. 謝辞

本研究に多大なるご支援と助言をいただいた神戸大学 ROOT プログラムの、相馬聡文教授と、NICT SecHack365 の、特に研究駆動コースのトレーナーとアシスタントの皆様に深く御礼申し上げます。

9. 図表

図 1

ブロッホ球

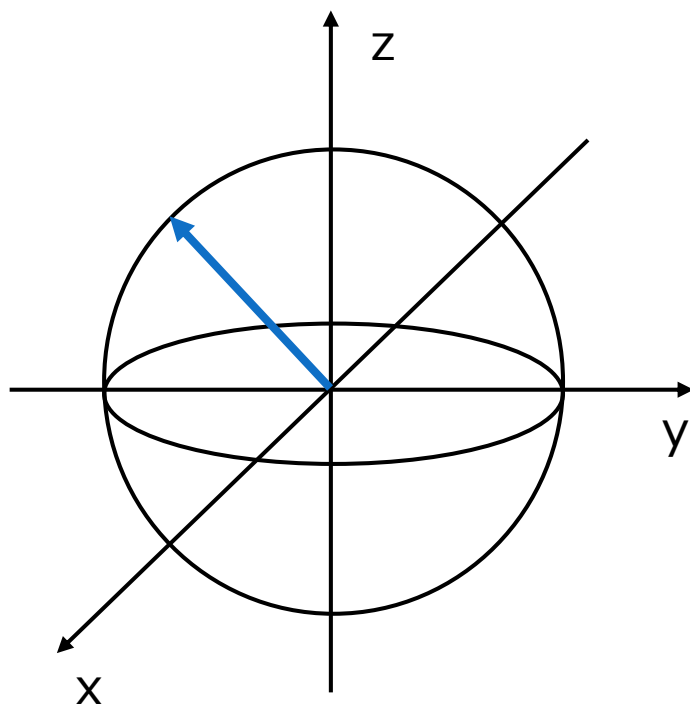


図 2

古典的ニューラルネットワークの略図

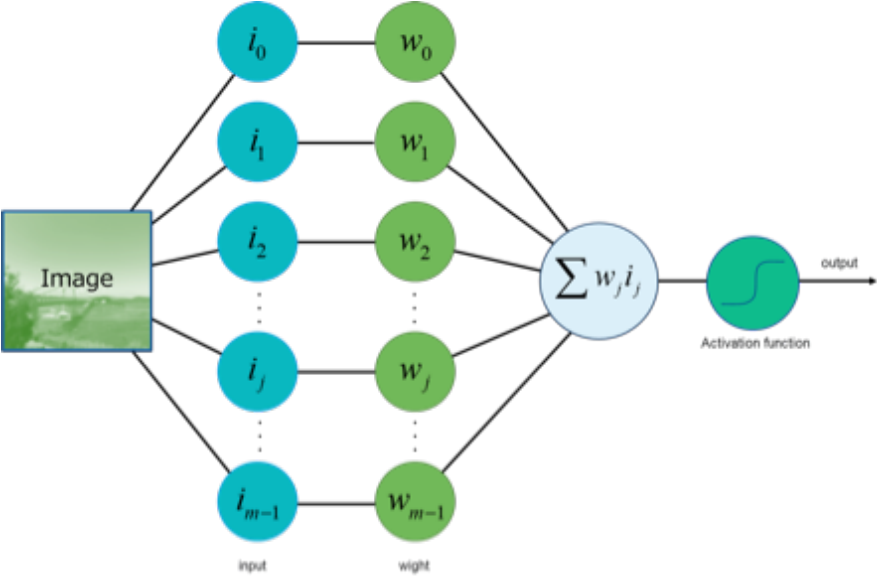


図 3

量子ニューラルネットワークの回路の略図

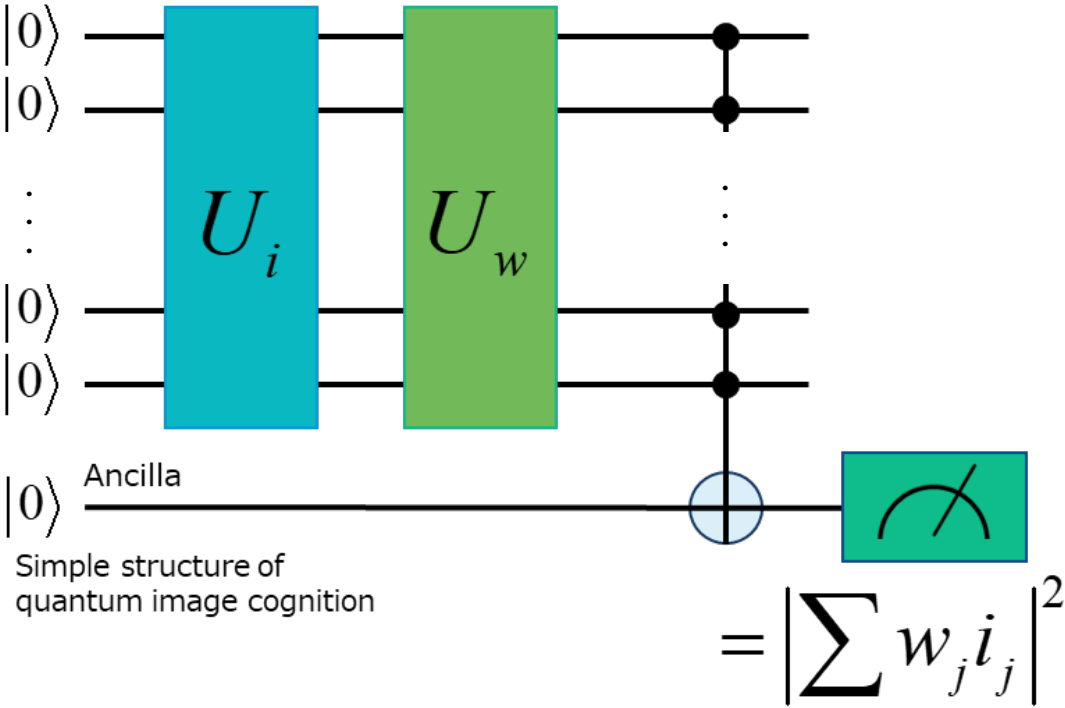


図 4

フェーズとグレースケールの対応関係

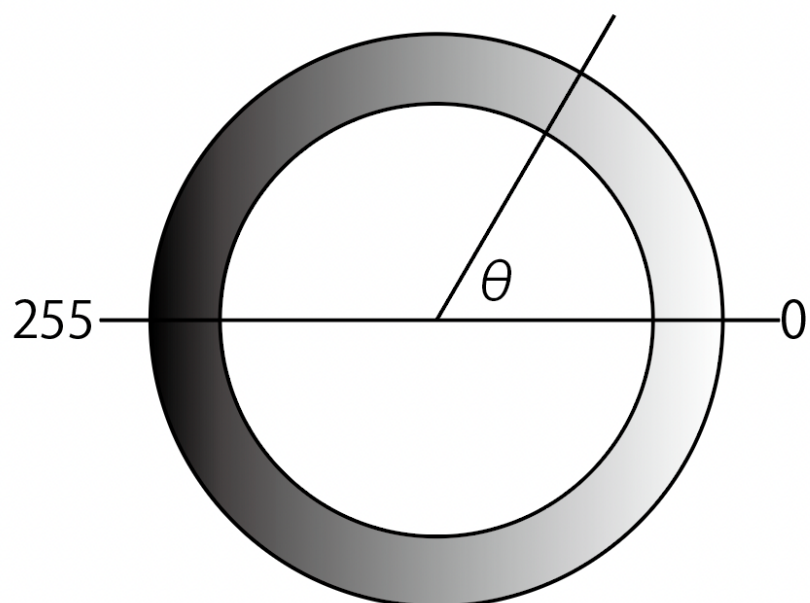
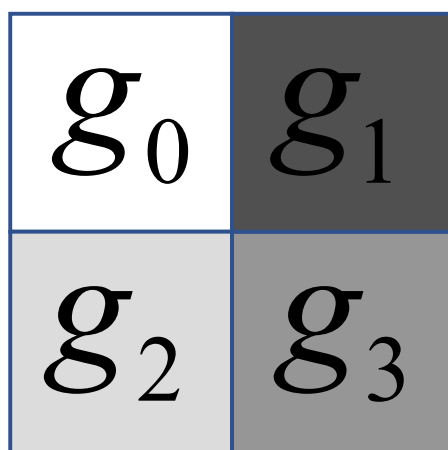


図 5

2×2 マスの白黒画像の例



$$0 \leq g_j < 256$$

図 6

CRZ ゲート

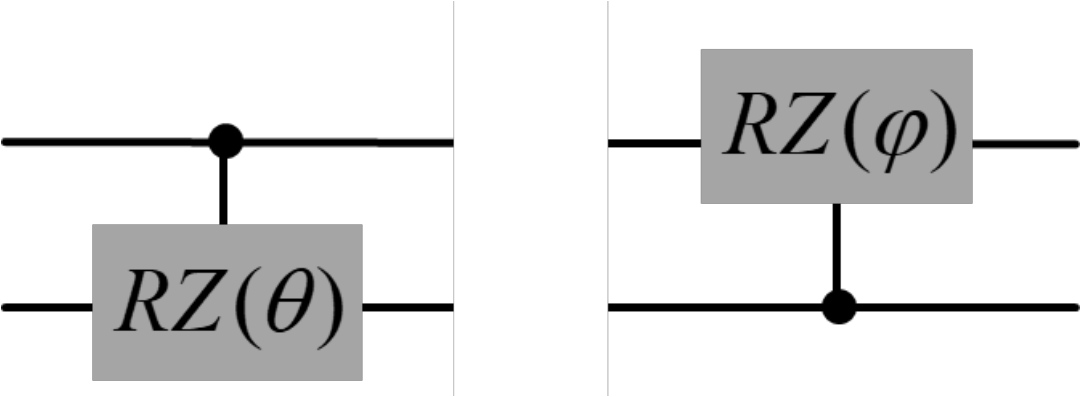


図 7

量子情報とその係数に収める画像情報の対応

000	000	000	000	000	000	000	000
000	001	010	011	100	101	110	111
001	001	001	001	001	001	001	001
000	001	010	011	100	101	110	111
010	010	010	010	010	010	010	010
000	001	010	011	100	101	110	111
011	011	011	011	011	011	011	011
000	001	010	011	100	101	110	111
100	100	100	100	100	100	100	100
000	001	010	011	100	101	110	111
101	101	101	101	101	101	101	101
000	001	010	011	100	101	110	111
110	110	110	110	110	110	110	110
000	001	010	011	100	101	110	111
111	111	111	111	111	111	111	111
000	001	010	011	100	101	110	111

図 8
 入力から Convolution 層、Pooling 層、観測の流れ

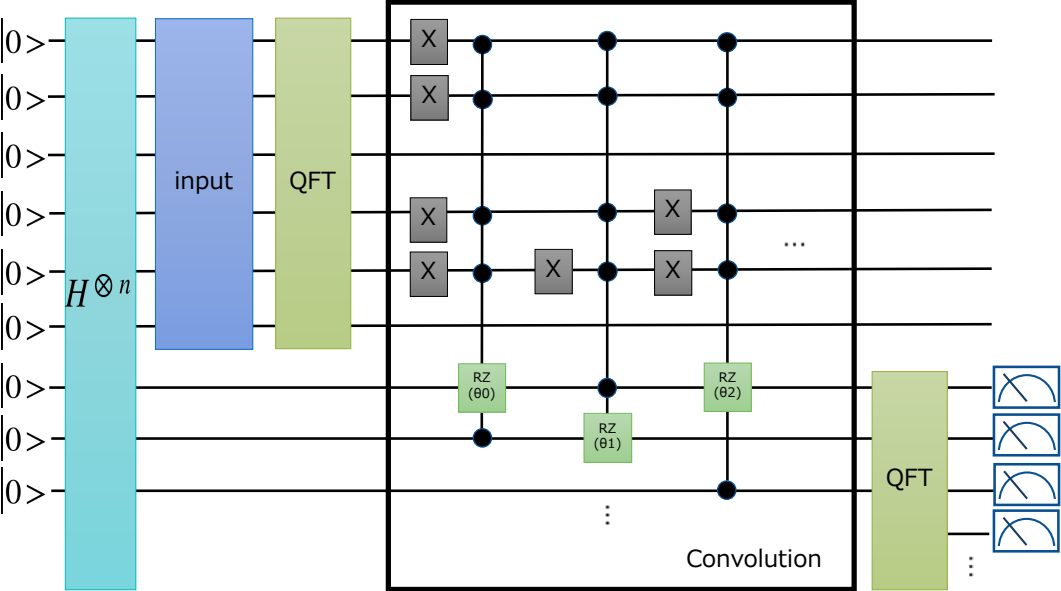


図 9
 畳み込みフィルタは格子を跨ぐことができない

000	000	000	000	000	000	000	000
000	001	010	011	100	101	110	111
001	001	001	001	001	001	001	001
000	001	010	011	100	101	110	111
010	010	010	010	010	010	010	010
000	001	010	011	100	101	110	111
011	011	011	011	011	011	011	011
000	001	010	011	100	101	110	111
100	100	100	100	100	100	100	100
000	001	010	011	100	101	110	111
101	101	101	101	101	101	101	101
000	001	010	011	100	101	110	111
110	110	110	110	110	110	110	110
000	001	010	011	100	101	110	111
111	111	111	111	111	111	111	111
000	001	010	011	100	101	110	111

図 10

複数のパディング方法による同じ画像の生成

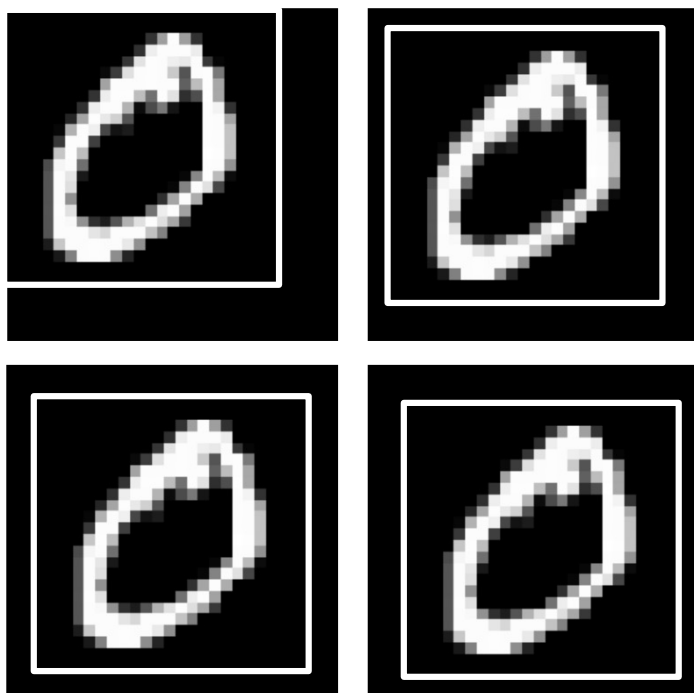


図 11

画像処理ステップ

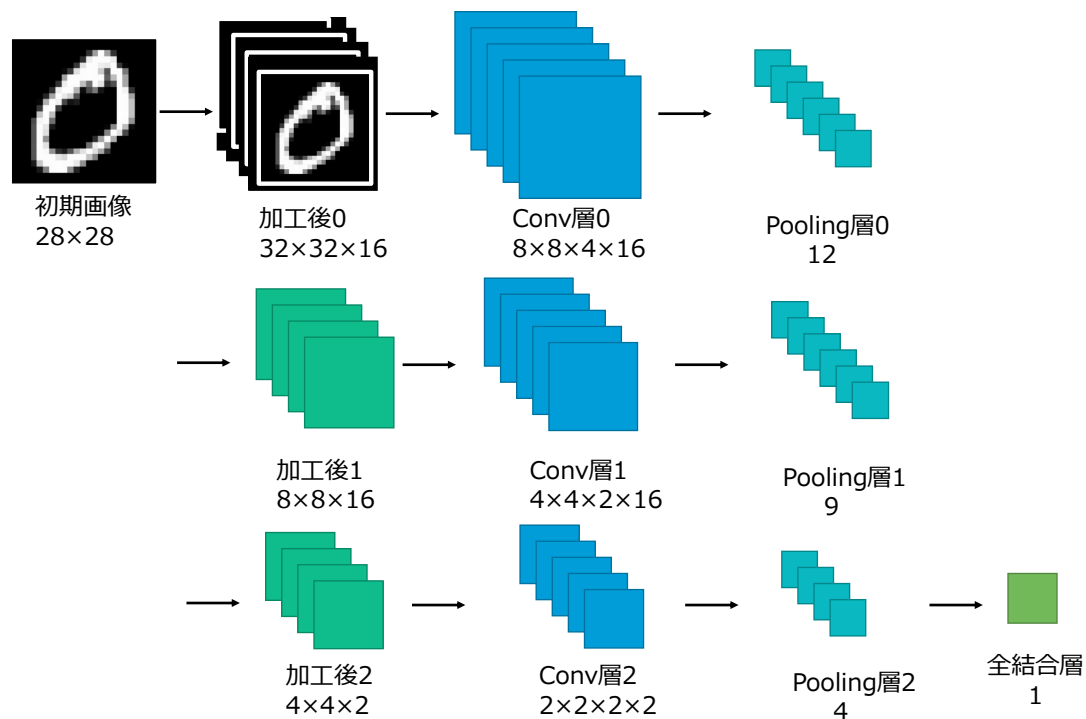


図 12

データサイズと層の数を減らした

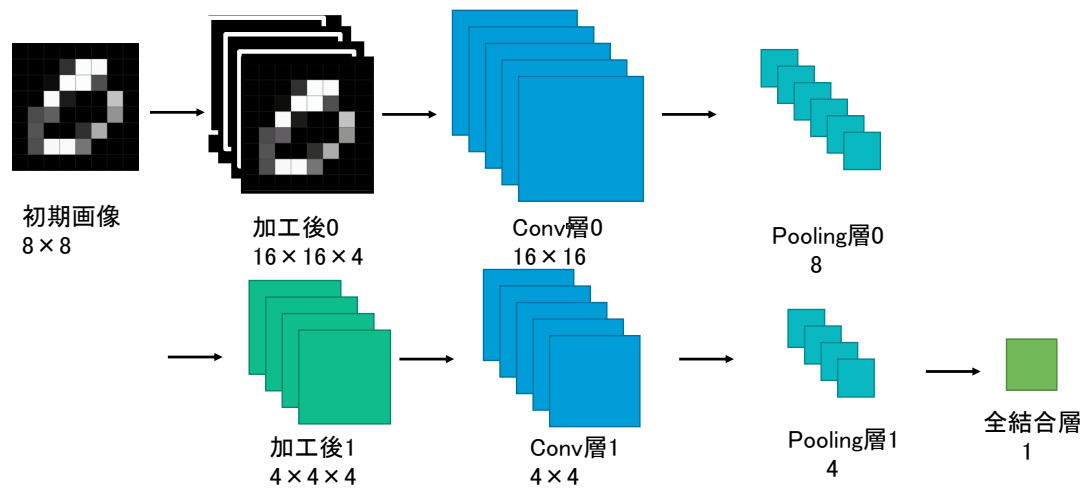


図 13

図 12 のモデルで損失関数は減少したが精度は向上しなかった。

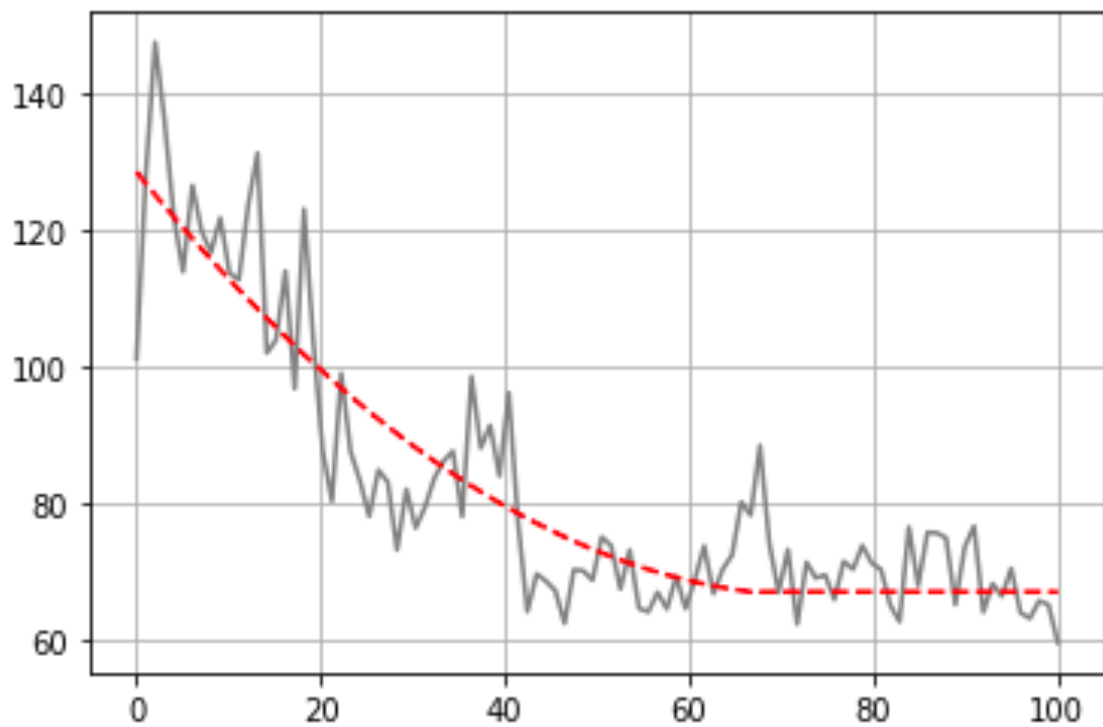


図 14

さらにデータの数減らし軽量化、パディングの欠点も削ったが精度は向上しなかった。

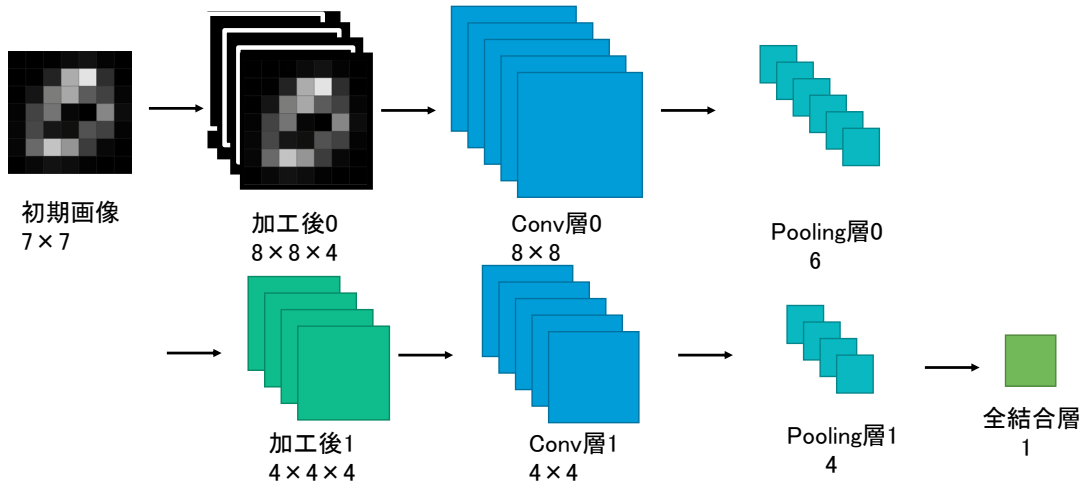


図 15

モデルの中に PyTorch を組み込み、一層目の出力結果を分類可能なデータへ変化させることを試みた。

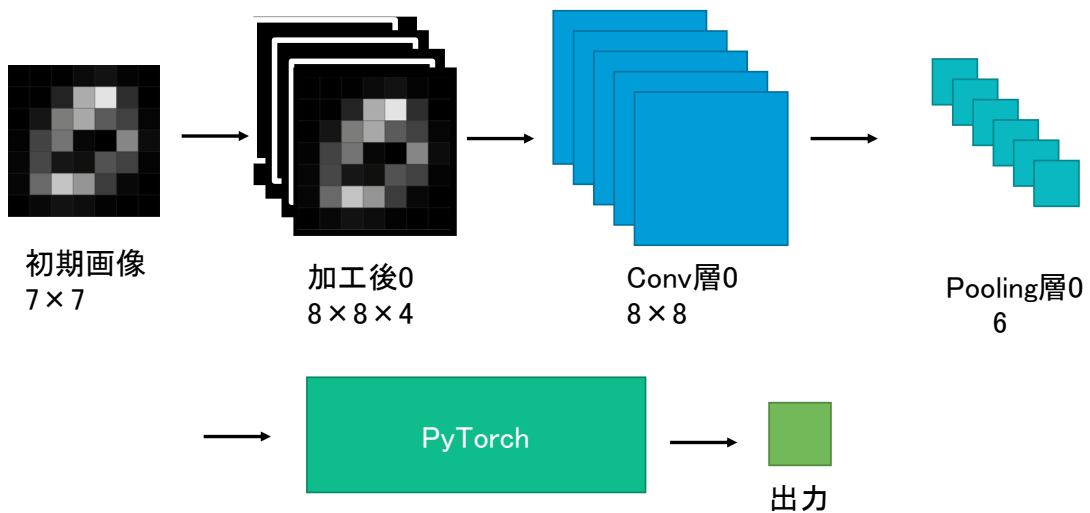
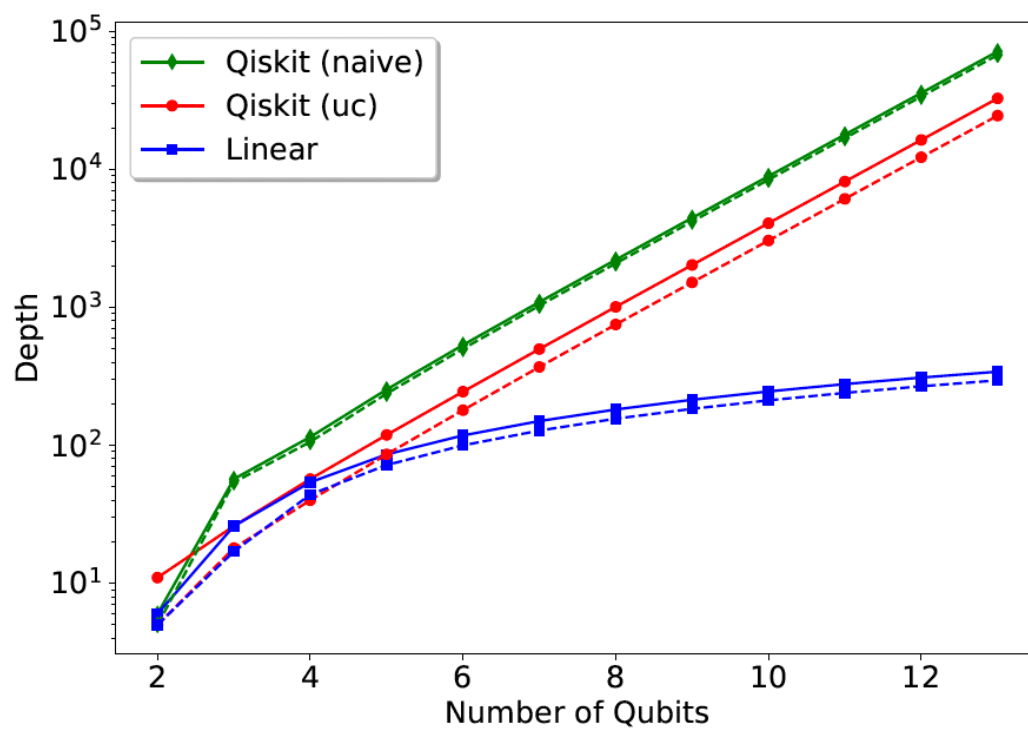


図 16

Qiskit で採用されている深さと Linear 法を用いた際の深さ (Da silva, A.j., & Park, D.K. (2022). Linear-Depth Quantum Circuits for Multi-Qubit Controlled Gates. Phys. Rev. A 106, 042602[10]より引用)



10. Appendix

同時摂動法を用いた微分の期待値は各変数における偏微分の値になっている。

$$f(\vec{x} + c\vec{s}) = f(\vec{x}) + c \sum_{i=1}^n \frac{\partial f(\vec{x})}{\partial x_i} s_i + \frac{c^2}{2} \sum_{i,j=1}^n \frac{\partial^2 f(\vec{x})}{\partial x_i \partial x_j} s_i s_j + \dots,$$

$$\bar{f} = f(\vec{x}), \nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix}, H = \nabla^2 \bar{f} = \begin{pmatrix} \frac{\partial^2 \bar{f}}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 \bar{f}}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \bar{f}}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 \bar{f}}{\partial x_n \partial x_n} \end{pmatrix},$$

$$f(\vec{x} + c\vec{s}) \simeq \bar{f} + c(\nabla \bar{f}, \vec{s}) + \frac{c^2}{2}(\vec{s}, H\vec{s}),$$

$$\frac{f(\vec{x} + c\vec{s}) - f(\vec{x})}{cs_i} \simeq (\nabla \bar{f}, \vec{s}) s_i + \frac{c}{2}(\vec{s}, H\vec{s}) s_i,$$

$$E\left(\frac{f(\vec{x} + c\vec{s}) - f(\vec{x})}{cs_i}\right) = \frac{\partial f(\vec{x})}{\partial x_i}, x_i' = x_i - \eta \frac{\partial f(\vec{x})}{\partial x_i}$$