

Continuous Variable Quantum MNIST Classifiers

—Classical-Quantum Hybrid Quantum Neural Networks

Sophie Choe, Marek Perkowski

Electrical and Computer Engineering, Portland State University, Portland, OR, USA

Email: sophchoe@gmail.com, h8mp@pdx.edu

How to cite this paper: Choe, S. and Perkowski, M. (2022) Continuous Variable Quantum MNIST Classifiers. *Journal of Quantum Information Science*, 12, 37-51. <https://doi.org/10.4236/jqis.2022.122005>

Received: April 6, 2022

Accepted: June 20, 2022

Published: June 23, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution-NonCommercial International License (CC BY-NC 4.0). <http://creativecommons.org/licenses/by-nc/4.0/>



Open Access

Abstract

In this paper, classical and continuous variable (CV) quantum neural network hybrid multi-classifiers are presented using the MNIST dataset. Currently available classifiers can classify only up to two classes. The proposed architecture allows networks to classify classes up to n^m classes, where n represents cutoff dimension and m the number of qumodes on photonic quantum computers. The combination of cutoff dimension and probability measurement method in the CV model allows a quantum circuit to produce output vectors of size n^m . They are then interpreted as one-hot encoded labels, padded with $n^m - 10$ zeros. The total of seven different classifiers is built using 2, 3, ..., 6, and 8-qumodes on photonic quantum computing simulators, based on the binary classifier architecture proposed in “Continuous variable quantum neural networks” [1]. They are composed of a classical feed-forward neural network, a quantum data encoding circuit, and a CV quantum neural network circuit. On a truncated MNIST dataset of 600 samples, a 4-qumode hybrid classifier achieves 100% training accuracy.

Keywords

Quantum Computing, Quantum Machine Learning, Quantum Neural Networks, Continuous Variable Quantum Computing, Photonic Quantum Computing, Classical Quantum Hybrid Model, Quantum MNIST Classification

1. Introduction

Unlike the original assumption that quantum computers would replace classical computers, quantum processing units (QPUs) are emerging as task-specific special purpose processing units much like Graphics Processing Units. The cur-

rently available QPUs are called near-term quantum devices, because they are not yet fully fault-tolerant and are characterized by shallow and short quantum circuits. Nonetheless, the availability of these devices allows for active research on quantum algorithms specific for them, especially in quantum chemistry, Gaussian boson sampling, graph optimization, and quantum machine learning.

The QPUs are based on two different theoretical models of quantum computing: the discrete variable (qubit-based) model and the continuous variable model [2]. The discrete variable model is an extension of the computational space from $\{0, 1\}$ to a two-dimensional complex space, with the computational basis $\{|0\rangle, |1\rangle\}$ [3]. The CV model is an extension of the computational space to an infinite dimensional Hilbert space whose computational basis is infinite, *i.e.*, $\{|0\rangle, |1\rangle, \dots, |n\rangle, \dots\}$ [2]. The quantum state of an information channel (qumode) is represented by an infinite complex projective linear combination of these basis states. In practice, it is approximated by a finite linear combination using only a finite basis $\{|0\rangle, |1\rangle, \dots, |n\rangle\}$. The number of basis states used for the approximation is called cutoff dimension. On an m -qumode system with cutoff dimension n , the computational state space is of dimension n^m . By varying cutoff dimension and the number of qumodes used for computation, we can control the dimension of the computational space.

Implementing machine learning algorithms on quantum computers is an active area of research. Quantum machine learning algorithms can be implemented on variational quantum circuits with parametrized quantum gates [4]. A quantum circuit is a collection of quantum gates and the change of states induced by the circuit of the initial quantum state is considered quantum computation. The results of quantum computing extracted via measurement are incorporated into optimization and parameter update computations performed on classical circuits [4].

Quantum neural networks (QNN), a subset of quantum machine learning, follow the same architectural framework: QNN on a QPU and optimization on a CPU. The main components of classical neural networks described as $L(x) = \phi(Wx + b)$ are the affine transformation $Wx + b$ and the nonlinear activation function $\phi(\cdot)$. In the qubit model, all the available unitary gates are linear in nature; hence, a direct implementation of bias addition and nonlinear activation function is not feasible. In the CV model, the displacement gate implements bias addition and the Kerr gate, nonlinear activation function. Therefore, naturally embedded in the model is a direct translation of $L(x) = \phi(Wx + b)$ into its quantum version $L(|x\rangle) = |\phi(Wx + b)\rangle$.

The proposed CV MNIST classifiers are built on Xanadu's X8 simulator, which simulates an 8-qumode photonic quantum computer. The ability to control the size of output vectors based on the number of qumodes and the notion of cutoff dimension allows for producing one-hot encoded labels of MNIST dataset. Different architectures on 2-qumodes, 3-qumodes, up to 8-qumodes are introduced. They are classical-quantum hybrid networks with classical feedforward neural networks, quantum data encoding circuits, and CV quantum neural network

circuits according to the CV binary classifier proposed by Killoran *et al.* [1]. The quantum machine learning software library, PennyLane [5], is used for the quantum circuit and Tensorflow Keras is used for the classical circuit and optimization. The classifiers achieve above 95% training accuracy.

This paper is organized in the following manner: in Section 2, the CV model of quantum computing is discussed, especially the infinite dimensionality of the computational state space and how the notion of cutoff dimension allows us to control the dimension of the space per computational instance. In Section 3, the details of the CV model of quantum neural networks are examined. In Section 4, the architecture of the seven hybrid classifiers and the experimental results are presented.

2. Continuous Variable Quantum Computing

The quantum state space of the CV model is an infinite dimensional Hilbert space, which is approximated by a finite dimensional space for practical information processing purposes. Under phase space representation of the model, the computational basis states are Gaussian states as opposed to single points. Therefore, the quantum gates taking these Gaussian states to Gaussian states offer richer arrays of transformations than just unitary (linear) transformations as in the qubit model.

The CV model is based on the wave-like property of nature. It uses the quantum state of the electromagnetic field of bosonic modes (qumodes) as information carriers [6]. Its physical implementation is done, using linear optics containing multiple qumodes [7]. Each qumode contains a certain number of photons probabilistically, which is manipulated with quantum gates for computation.

A photon is a form of electromagnetic radiation. The position wave function $\Psi(x)$ describes the light wave electromagnetic strength of the qumode, depending on the number of photons present in it, expressed as

$\Psi : \mathbb{R} \rightarrow \mathbb{C} : x \mapsto \alpha$, where x is the position(s) of the photon(s). It is a complex valued function on real valued variables. The wave function with more than one photon displays the constructive and destructive interactions between the light waves of the photons.

Phase space representation of the quantum state of a linear optical system describes the state using the position x and momentum p variables of the photons in the given qumode system. The quasi-probability distribution of x and p are plotted on the xp -plane, using the Wigner function. It is given by

$$W(x, p) = \frac{p}{h} = \frac{1}{h} \int_{-\infty}^{\infty} e^{-\frac{ipy}{h}} \Psi\left(x + \frac{y}{2}\right) \Psi^*\left(x - \frac{y}{2}\right) dy \quad (1)$$

where $h = 6.62607015 \times 10^{-34}$ is the Plank constant and $\hbar = 6.582119569 \times 10^{-16}$ the reduced Plank constant [8].

This Wigner function is applied to the position wave function $\Psi_k(x)$, where k denotes the number of photons present in a qumode, to create Fock basis, also

known as number basis. The image of the Wigner functions

$W_0(x_0, p_0), W_1(x_1, p_1), \dots, W_4(x_4, p_4)$ where x_k and p_k represent the position and momentum variables with k photons present in the system is shown in

Figure 1.

They are used as computational basis states expressing the quantum state of a system. The quantum state $|\psi\rangle$ of a qumode is expressed as a superposition of Fock basis states:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_n|n\rangle + \dots, \text{ where } \sum_{k=0}^{\infty} \|c_k\|^2 = 1. \quad (2)$$

In practice, there are not going to be an infinite number of photons physically present in a qumode. We approximate $|\psi\rangle$ with $|\hat{\psi}\rangle$ by cutting off the trailing terms. The number of Fock basis states we use to approximate the true state $|\psi\rangle$ is called ‘‘cutoff dimension’’. Let n be cutoff dimension. Then the approximating state $|\hat{\psi}\rangle$ is in superposition of $|0\rangle, |1\rangle, \dots, |n-1\rangle$, *i.e.*,

$$|\hat{\psi}\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_{n-1}|n-1\rangle, \text{ where } \sum_{k=0}^{n-1} \|c_k\|^2 = 1. \quad (3)$$

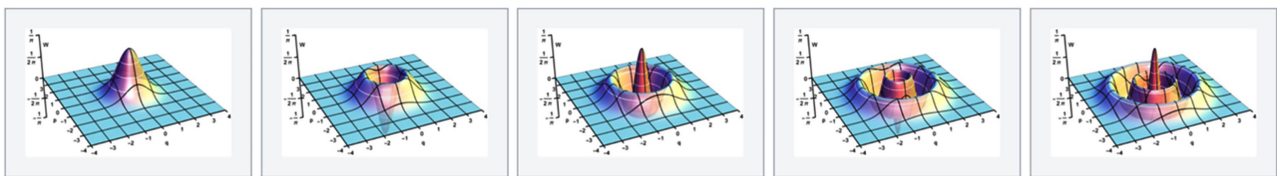
In vector representation, the state is expressed as a column vector of size n = cutoff dimension, with the k^{th} entry being c_k .

A multi-qumode system is represented by the tensor product of individual qumode states: $|\psi_0\rangle \otimes |\psi_1\rangle \otimes \dots \otimes |\psi_{m-1}\rangle$, where m is the number of qumodes. Where these states are approximated by cutoff dimension n , the computational basis of the resulting quantum state space is of size n^m . Then the approximated state is expressed as

$$|\hat{\psi}_0\rangle \otimes |\hat{\psi}_1\rangle \otimes \dots \otimes |\hat{\psi}_{m-1}\rangle = d_0|00\dots 0\rangle + d_1|00\dots 1\rangle + \dots + c_{n^m-1}|n-1, n-1, \dots, n-1\rangle. \quad (4)$$

Fock basis states constitute the eigenstates of the number operator $\hat{n} := \hat{a}^\dagger \hat{a}$, where \hat{a}^\dagger is called the constructor and \hat{a} the annihilator [2]. Their matrix representation is given by

$$\hat{a}^\dagger = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \sqrt{1} & 0 & 0 & \dots & 0 & 0 \\ 0 & \sqrt{2} & 0 & \dots & 0 & 0 \\ 0 & 0 & \sqrt{3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \sqrt{n-1} & 0 \end{bmatrix} \text{ and } \hat{a} = \begin{bmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sqrt{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (5)$$



Wigner function of $|0\rangle$ Wigner function of $|1\rangle$ Wigner function of $|2\rangle$ Wigner function of $|3\rangle$ Wigner function of $|4\rangle$

Figure 1. Fock basis.

The constructor $\hat{a}^\dagger |k\rangle = \sqrt{k+1} |k+1\rangle$ for $k \geq 0$ indeed constructs the next level Fock basis state and the annihilator $\hat{a} |0\rangle = 0$, $\hat{a} |k\rangle = \sqrt{k-1} |k-1\rangle$ for $k > 0$ annihilates the current level Fock basis state to the lower.

The product of \hat{a}^\dagger and \hat{a} returns a matrix

$$\hat{a}^\dagger \hat{a} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & n-2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & n-1 \end{bmatrix} \tag{6}$$

which we denote as “number operator” \hat{n} . The Fock basis states $|k\rangle$ are indeed eigenstates of the number operator as shown in $\hat{n} |k\rangle = k |k\rangle$, where $k \in \{0, 1, \dots, n-1\}$ for cutoff dimension = n .

Standard Gaussian gates in the CV model are of the form $U = e^{-iHt}$, where H represents the Hamiltonian of the system, describing the total energy as the sum of kinetic and potential energy. The matrix exponential of U is given by

$$U = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!} = I - iHt + \frac{(-iHt)^2}{2!} + \cdots + \frac{(-iHt)^n}{n!} + \cdots \tag{7}$$

Some of the standard Gaussian gates taking Gaussian states to Gaussian states are:

Squeezer with parameter z : $S(z) = \exp\left(\frac{z^* \hat{a}^2 - z \hat{a}^{\dagger 2}}{2}\right)$

Rotation with parameter ϕ : $R(\phi) = \exp(i\phi \hat{a}^\dagger \hat{a})$

Displacement with parameter α : $D(\alpha) = \exp(\alpha \hat{a}^\dagger - \alpha^* \hat{a})$

Beamsplitter, a two-qumode gate, with parameters θ and ϕ :

$$B(\theta, \phi) = \exp\left(\theta \left(e^{i\phi} \hat{a} \hat{b}^\dagger - e^{-i\phi} \hat{a}^\dagger \hat{b} \right)\right),$$

where \hat{b}^\dagger and \hat{b} are constructor and annihilator of the second qumode respectively.

Measurement is done via counting the number of photons present in each qumode with a photon detector. Xanadu’s PennyLane offers a suite of measurement methods as outlined in **Table 1**.

Table 1. Xanadu PennyLane measurement methods.

Measurement Method	Output Size	
	Per Qumode	For m -qumodes
Expectation Value	1	m
Variance	1	m
Probability	n	n^m

The size of the output vectors is affected by cutoff dimension n and the number of qumodes used for computation. Currently available QPU by Xanadu, X8, offers eight qumodes. The measurement methods used in the proposed MNIST classifiers are the expectation value method and the probability method.

The expectation value method produces a single real valued output. Let $|\psi\rangle = \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix}$ be the quantum state of a multi-qumode system after desired quantum computational operations are performed. The expectation value measurement method returns $\langle\psi|X|\psi\rangle$, where the operator A is usually the Pauli- X , Pauli- Y , or Pauli- Z gate. The expectation value of the Pauli- X matrix is

$$\begin{aligned} \langle\psi|X|\psi\rangle &= \begin{bmatrix} \psi_0^* & \psi_1^* \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix} \\ &= \begin{bmatrix} \psi_0^* & \psi_1^* \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_0 \end{bmatrix} \\ &= \psi_0^* \psi_1 + \psi_1^* \psi_0 \\ &= 2[\operatorname{Re}(\psi_0)\operatorname{Re}(\psi_1) + \operatorname{Im}(\psi_0)\operatorname{Im}(\psi_1)] \in \mathbb{R}, \end{aligned} \quad (8)$$

which is a real number. In a multi-qumode system with m qumodes, we can get an output vector of size m by getting the expectation value for each qumode.

The probability method returns a vector, with each entry indicating the probability of getting the corresponding computational basis state. Suppose an m -qumode system has cutoff dimension n for each qumode. Then, there are n^m computational basis states, hence we get a vector of size n^m .

3. Quantum Neural Networks

Machine learning is a way of extracting hidden patterns from data by learning a set of optimal parameters for a mathematical expression that most closely match the data. The mathematical expression used for pattern extraction is called a machine learning algorithm. An algorithm with an optimal set of parameters, learned via training, is called a model. With near-term devices available on cloud, execution of quantum machine learning (QML) algorithms on quantum computers or simulators is now feasible.

QML algorithms are built with variational circuits, *i.e.*, parameterized circuits, composed of quantum gates whose actions are defined by parameters. Training is the process of “learning” optimal parameters of the gates, which produce as accurate inferences as possible for new data samples. The measurement results from a variational circuit run on a QPU are sent to a CPU for parameter optimization, *i.e.*, computation of objective function, gradients, and new parameters. The updated parameters are fed back to the quantum circuit to adjust the parameterized gates for the subsequent iteration. The illustration of the process is shown in **Figure 2**. Google and Xanadu offer Python based software packages specifically for quantum machine learning: Tensorflow Quantum (Google) and PennyLane (Xanadu) [5].

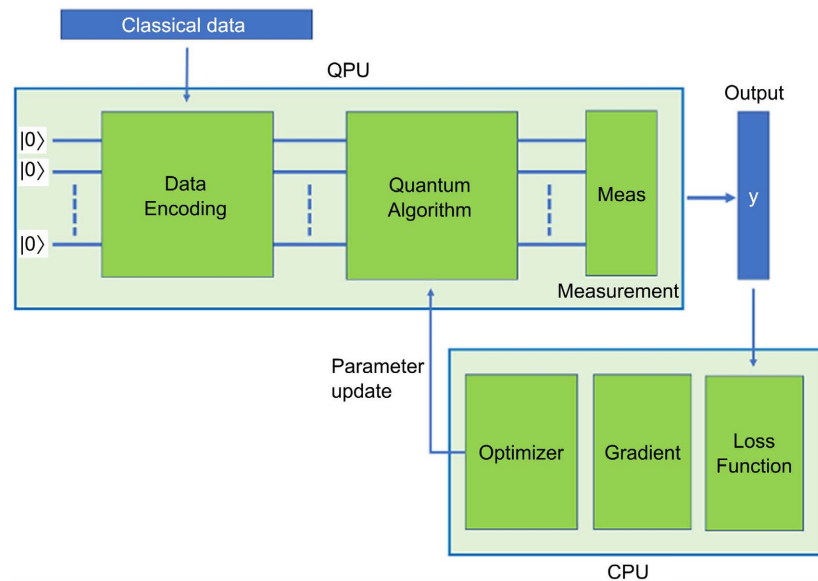


Figure 2. Variational quantum circuit parameter update.

Neural network is one of the subsets of machine learning algorithms, defined by a stack of layers, each composed of an affine transformation $Wx+b$ and a nonlinear activation function $\phi(\cdot)$. Each layer of a neural network can be mathematically described as $L(x) = \phi(Wx+b)$. The output from one layer is then fed as input into the subsequent layer. The entire network is a composition of different layers: $L(x) = L_m \circ L_{m-1} \circ \dots \circ L_1(x)$. The entries of the matrix W and the bias vector b for each layer are learned as parameters through an iterative training process given an objective function. The goal is to find an optimal set of parameters $\{W_1, W_2, \dots, W_m, b_1, b_2, \dots, b_m\}$ for a network of m layers.

In quantum neural networks, the objective is to implement the classical mathematical expression $L(x) = \phi(Wx+b)$ as a quantum state $L(|x\rangle) = |\phi(Wx+b)\rangle$. In converting classical neural networks into quantum circuits, the key components are:

- Data encoding: $x \mapsto |\psi(x)\rangle$
- Affine transformation: $W|\psi(x)\rangle + |b\rangle$
- Nonlinear activation function: $\phi(|\cdot\rangle)$.

In the qubit model, all available unitary gates are linear. Hence a direct way of implementing the bias addition and nonlinear activation function components in quantum is absent in the model.

3.1. Continuous Variable Quantum Neural Networks

Naturally embedded in the CV model are quantum gates, which allow for direct implementations of the expression $L(|x\rangle) = |\phi(Wx+b)\rangle$. The affine transformation $Wx+b$ is implemented by the composition $D \circ U_2 \circ S \circ U_1$, where U_k denotes the k^{th} interferometer, S a set of m squeezers, D a set of m displacement gates. The activation function $\phi(|\cdot\rangle)$ is implemented by a set of Kerr gates, which are nonlinear. The composition $\phi \circ D \circ U_2 \circ S \circ U_1$ acting on a quantum

state $|x\rangle$ gives us the desired state $L(|x\rangle) = |\phi(Wx + b)\rangle$. The schematic of the circuit is shown in **Figure 3**. The interferometer U_k on an m -qumode system is composed of $m - 1$ beam splitters and m rotation gates as shown in **Figure 4**.

The action of a phase-less interferometer U_k on the quantum state $|x\rangle = \otimes_{k=1}^m |x_k\rangle$ has an effect of an orthogonal matrix action on $|x\rangle$ [1]. Orthogonal matrices are just unitary matrices with real valued entries, inducing length-preserving rotations. Then the transpose of an orthogonal matrix represents the reverse rotation of the original matrix, thus orthogonal. The composition $U_2 \circ S \circ U_1$ can be considered as the composition $O_2 \circ S \circ (O_1^T)^T$, where O_2 and O_1^T are orthogonal.

Let W be the linear transformation matrix we want to implement with a quantum circuit. Any matrix W can be factorized using Singular Value Decomposition as $W = U\Sigma V^*$, where U and V^* are orthogonal and Σ is diagonal [9]. The parameterized squeezer $S(r_k)$ acts on the quantum state $|x_k\rangle$ as $S(r_k)|x_k\rangle = \sqrt{e^{-r_k}} |e^{-r_k} x_k\rangle$ for each k^{th} qumode. Collectively they have an effect of a diagonal matrix $S = S_1 \otimes S_2 \otimes \dots \otimes S_m$ acting on $|x\rangle = \otimes_{k=1}^m |x_k\rangle$. The composition $U_2 \circ S \circ U_1$ implements a quantum version of the linear transformation matrix W [1].

The bias addition is realized with displacement gates D . The displacement gate has an effect $D(\alpha_k)|\psi_k\rangle = |\psi_k + \sqrt{2}\alpha_k\rangle$ for each k^{th} qumode. Then $D(\alpha)|\psi\rangle = |\psi + \sqrt{2}\alpha\rangle$ collectively for $\alpha^T = [\alpha_1, \alpha_2, \dots, \alpha_m]$. For some desired bias b , let $\alpha = \frac{b}{\sqrt{2}}$, then the collection of displacement gates implements the bias addition. The composition $D \circ U_2 \circ S \circ U_1$ acting on the quantum state $|x\rangle$ gives us the affine transformation

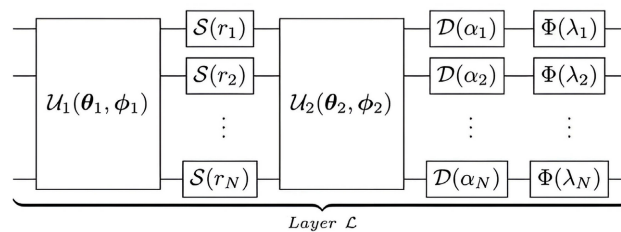


Figure 3. CV quantum neural network architecture [1].

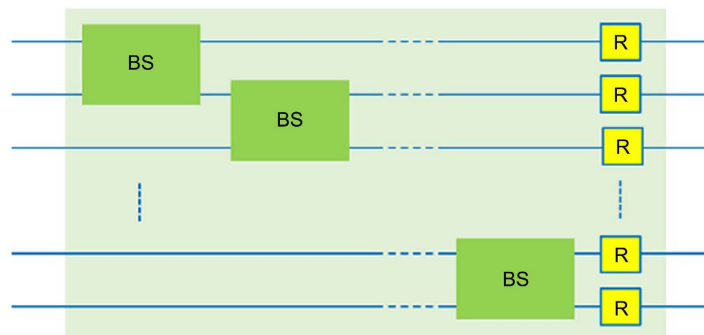


Figure 4. Interferometer architecture.

$$D \circ U_2 \circ S \circ U_1 |x\rangle = |O_2 \Sigma O_1 x + b\rangle = |Wx + b\rangle. \tag{9}$$

The nonlinear activation function $\phi(\cdot)$ is realized with a set of Kerr gates, that we denote Φ . The Kerr gate, parameterized by the parameter κ , is a nonlinear transformation gate. Let n be cutoff dimension and m be the number of qumodes. For the quantum state $|\psi\rangle$ of one qumode, which is a superposition of n Fock basis states, the Kerr gate with parameter κ has an effect

$$K(\kappa)|\psi\rangle = \begin{bmatrix} e^{i0^2\kappa} & 0 & 0 & \dots & 0 \\ 0 & e^{i1^2\kappa} & 0 & \dots & 0 \\ 0 & 0 & e^{i2^2\kappa} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & e^{i(N-1)^2\kappa} \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{N-1} \end{bmatrix} = \begin{bmatrix} \psi_0 \\ e^{i\kappa}\psi_1 \\ e^{i2^2\kappa}\psi_2 \\ \vdots \\ e^{i(N-1)^2\kappa}\psi_{N-1} \end{bmatrix} \tag{10}$$

which is nonlinear.

Together, the circuit $L = \Phi \circ D \circ U_2 \circ S \circ U_1$ gives us a quantum version $L(|x\rangle) = |\phi(Wx + b)\rangle$ of a classical neural network $L(x) = \phi(Wx + b)$.

3.2. Continuous Variable Binary Classifier

The binary classifier outlined in Continuous-variable quantum neural networks: is a classical and quantum hybrid network [1].

The dataset used contains 284,806 genuine and fraudulent credit card transactions with 29 features, out of which only 492 are fraudulent. The dataset is truncated to 10 features as per the paper and 1968 samples with 1:3 ration of fraudulent vs. genuine.

The classical-quantum hybrid model from the paper has a classical neural network taking input vectors of size 10 and outputting vectors of size 14, a quantum encoding circuit, and a 2-qumode quantum neural network which outputs vectors of size 2. We can regard the output vectors as one-hot encoding of binary classification between fraudulent vs. genuine. The architecture of the hybrid network is shown in Figure 5.

The dataflow of the circuit is

- Classical network: 2 hidden layers with 10 neurons, each using Exponential Linear Units (ELU) as activation function. Output layer with 14 neurons.

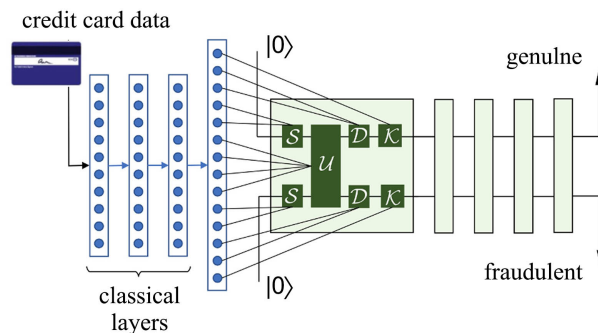


Figure 5. Binary hybrid classifier circuit [1].

- Data encoding: Output vector from the classical network is converted to a quantum state by the circuit composed of squeezers, interferometer, displacement gates, and Kerr gates.
- Quantum network: 4 layers of quantum neural network. Each layer is composed of interferometer 1, squeezers, interferometer 2, displacement gates, and Kerr gates.
- Measurement: The expectation value of the Pauli-X gate $\langle \psi_k | X | \psi_k \rangle$ is evaluated for each k^{th} qumode state $|\psi_k\rangle$.

The Keras-PennyLane implemented experiment yields 97% training accuracy.

4. Continuous Variable MNIST Classifiers

The multi-classifier models presented in this section are inspired by the CV binary classifier architecture described in Section 3.2 [1]. They are run on Xanadu's X8 photonic quantum simulator, made up of 8-qumodes. There are two models classifying eight classes and five models classifying ten classes. The measurement output vectors from these models are interpreted as one-hot encoded predictions of the image labels.

The classifiers on eight classes are built using 3-qumodes and 8-qumodes, because of the output size we can get from those structures with the use of different measurement methods as shown in **Table 2**.

A classical neural network is used as a pre-processing step to reduce the original image matrices of size $28 \times 28 = 784$ to vectors of lower length that the data encoding circuit accommodates. For the data encoding circuit, squeezers, interferometers, displacement gates, and Kerr gates are used. For the quantum neural network, the quantum circuit implementing $L(|x\rangle) = |\phi(Wx + b)\rangle$ as in the binary classifier is used.

The 10-class classifiers are built on 2, 3, ..., 6-qumodes. The label $k \in \{0, 1, \dots, 9\}$ of an image matrix, when converted into a one-hot encoded vector, becomes a vector of length 10 with all zeros but the k^{th} entry as 1. The cutoff dimensions are selected so that the output vectors exceed 10 in length. For each classifier, a different number of zeros are padded to the one-hot encoded labels to match the output size of the circuit. The cutoff dimension used for each classifier and the size of output vectors are shown in **Table 3**.

The architecture in **Figure 6** depicts the dataflow: image matrix \rightarrow classical layers \rightarrow reduced output vector \rightarrow data encoding \rightarrow QNN \rightarrow measurement \rightarrow output vector as a one-hot encoded label.

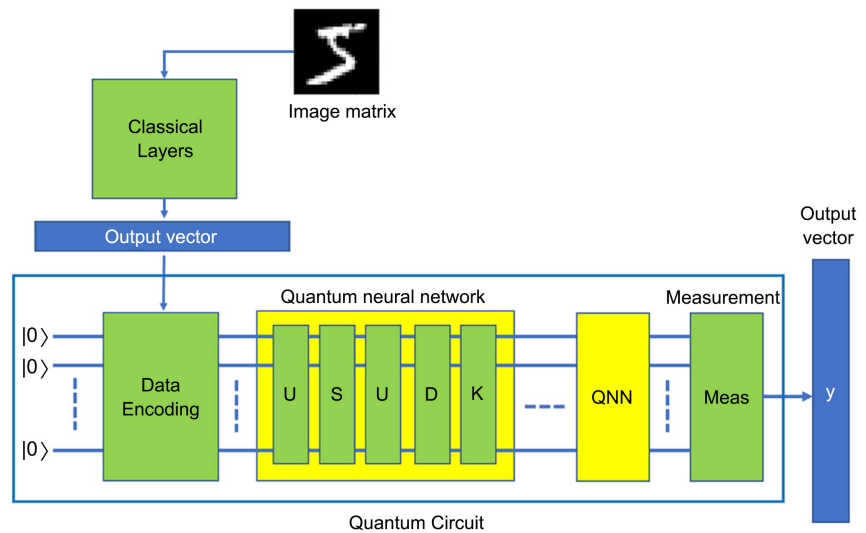
The boxes U , S , D , and K represent interferometers, a set of m squeezers, a set of m displacement gates, and a set of m Kerr gates respectively, where $m =$ the number of qumodes.

Table 2. Measurement methods used for eight-class classifiers.

Number of Qumodes	Cutoff Dimension	Measurement Method	Output Size
3	2	Probability	$2^3 = 8$
8	2	Expectation Value	$1 \times 8 = 8$

Table 3. Output size and the number of zero paddings for ten-class classifiers.

Number of Qumodes	Cutoff Dimension	Output Size	Number of Zero Paddings
2	4	$4^2 = 16$	6
3	3	$3^3 = 27$	17
4	2	$2^4 = 16$	6
5	2	$2^5 = 32$	22
6	2	$2^6 = 64$	54

**Figure 6.** MNIST multi-classifier architecture.

4.1. Classical Layers

Classical feedforward neural networks are used for preprocessing of data images to reduce the size of $28 \times 28 = 784$ to smaller size vectors fitting the number of parameters available for data encoding. The image matrices are flattened to vectors of length 784 and then reduced to vectors of smaller length through Keras dense layer operations with activation function ELU. The output vectors are then encoded as quantum states by the data encoding quantum circuit.

4.2. Data Encoding

The output vectors from the classical neural network are in classical states. The quantum data encoding circuit converts classical states into quantum states. The quantum gates used for data encoding are squeezers, an interferometer, displacement gates, and Kerr gates. The entries of a classical vector are used as the parameters of these parameterized quantum gates.

An interferometer is composed of beam splitters on pairs of adjacent qumodes and rotation matrices. Squeezers $S(z)$ and displacement gates $D(\alpha)$ can be considered either one-parameter gates or two-parameter gates when the parameters $z, \alpha \in \mathbb{C}$ are converted in Euler formula

$z = a + bi = r(\cos \theta + i \sin \theta) = re^{i\theta}$. In the proposed circuit, they are used as two-

parameter gates.

The number of parameters that these gates can accommodate for m -qumode circuits are $8m - 2$, as shown in **Table 4**.

Based on these values, the size of the classical network output vectors was determined.

4.3. Quantum Circuit

The QNN circuit implements a quantum version of the classical neural network $L(x) = \phi(Wx + b)$ as $L(|x\rangle) = |\phi(Wx + b)\rangle = \Phi \circ D \circ U_2 \circ S \circ U_1 |x\rangle$. The number of parameters per collection of gates on m -qumodes is shown in **Table 5**.

On the 8-qumode classifier, two layers of QNN are applied. On the rest of the classifiers, four layers are applied. The number of parameters that the network is designed to learn for the varying numbers of qumodes are shown in **Table 6**.

4.4. Measurement

Pennylane offers three different measurement methods for CV-based computations: expectation value, variance, and probabilities. The expectation value and variance methods yield a single-valued result for each qumode. The probability method yields vectors of size n^m , where n = cutoff dimension and m = the number of qumodes.

For the eight-qumode model, the expectation value method was used. The result is a vector of length eight, $[\langle \psi_0 | X | \psi_0 \rangle, \langle \psi_1 | X | \psi_1 \rangle, \dots, \langle \psi_7 | X | \psi_7 \rangle]$, where $\langle \psi_k | X | \psi_k \rangle$ represents the expectation value measurement method of the $(k + 1)^{\text{th}}$ qumode. It is then interpreted as a one-hot encoded label vector of the corresponding image.

For the rest of the models, the probability method was used.

Table 4. Number of parameters per a set of gaussian gates.

Squeezer	Beamsplitter	Rotation	Displacement	Kerr
$2m$	$2(m - 1)$	m	$2m$	m

Table 5. Formula for the number of parameters for an m -qumode system.

Total	$U_k \times 2$	S	D	$\phi(\cdot)$
$2 \cdot 2(m - 1) + 5m = 9m - 4$	$2(2(m - 1) + m)$	m	m	m

Table 6. Number of parameters per number of qumodes.

Number of Qumodes	Per Layer	Total
2	14	56
3	23	92
4	32	128
5	41	164
6	50	200
8	68	136

4.5. Parameter Update

With the PennyLane Tensorflow plug-in functionality, the quantum circuit is converted a Keras layer and added to the classical layers. Then Keras' built-in loss functions and optimizers can be used for parameter update. For most of the models, Categorical Crossentropy is used for loss function and Stochastic Gradient Descent is used for optimizer. For the 8-qumode classifier, the Mean Squared Error loss function performed better. The updated parameters are then used as the parameters of the quantum gates for the subsequent iteration of training.

4.6. Experimental Results

The 4-qumode classifier yielded the best result of 100% training accuracy on 600 data samples. Accuracy comparison with the qubit-based binary classifiers using Tensorflow Quantum and Qiskit is listed in **Table 7**.

All the classifiers tested achieve above 95% training accuracy. For the 8-qumode classifier, 300 samples and two layers of QNN were used with 50 epochs. For the rest of the classifiers, 600 samples and four layers of QNN were used. With the 4-qumode classifier, training accuracy of 100% is achieved in 70 epochs.

The loss and accuracy for the models with varying numbers of qumodes are listed in **Table 8**.

All of these classifiers followed the typical loss and accuracy graphs depicted in **Figure 7**. The validation accuracy is not as high as the training accuracy, indicating "over-fitting". Further experiments with different hyper-parameters or the use of regularizer are called for.

Table 7. Comparison of qubit-based classifiers and CV classifiers.

	Tensorflow Quantum	Qiskit-Pytorch	PennyLane-Keras
Number of Classes	2	2	10
Number of Samples	10,338	100	600
Training Accuracy	89.92%	100%	100%

Table 8. Table type styles (Table caption is indispensable).

Number of Qumodes	Learning Rate	Training Loss	Training Accuracy
2	0.02	0.4966	97.14%
3	0.05	0.0563	99.90%
3*	0.05	0.4567	98.04%
4	0.03	0.1199	100.00%
5	0.02	0.2057	98.40%
6	0.02	0.2055	98.22%
8	0.1	0.0199	99.77%

*The 3-qumode classifier is an eight-class classifier.

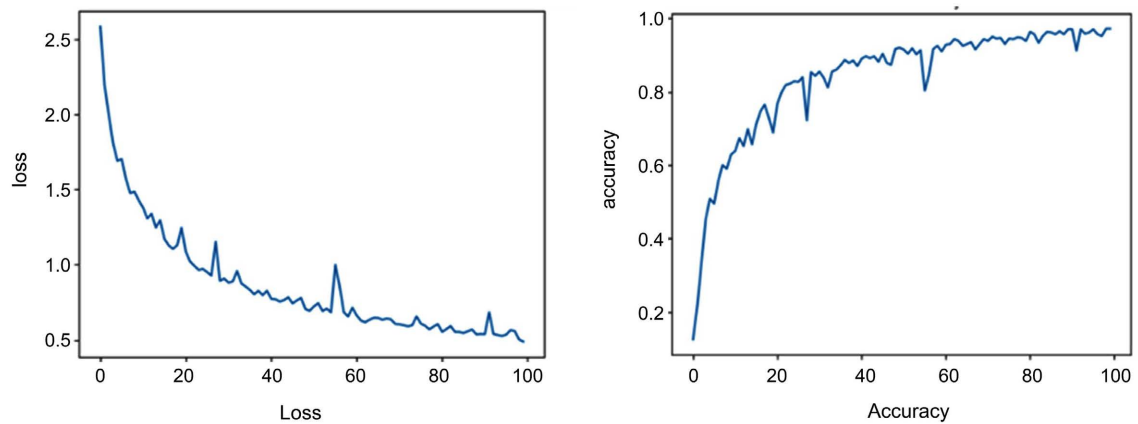


Figure 7. MNIST multi-classifier experimental results.

5. Conclusions

In this paper, classical and CV quantum hybrid classifiers using different numbers of qumodes and cutoff dimensions were examined. The contribution of this paper is proposing and successfully implementing quantum neural network classifiers with more than two classes. Currently available quantum neural network architectures can only classify up to two classes.

The quantum gates available in the CV model allow a natural implementation of a quantum neural network layer $L(|x\rangle) = |\phi(Wx + b)\rangle$. The flexibility of different measurement methods to output vectors of different lengths allows the networks to produce results that are interpreted as one-hot encoded labels. Therefore, any CV QNN with the proposed architecture with cutoff dimension n and m -qumodes is capable of classifying up to n^m classes.

There is a limitation in encoding classical data into quantum states in near-term devices due to the number of qumodes, which is currently eight on Xanadu's X8 photonic QPU. Classical networks were used to reduce the number of entries in the image matrices for quantum data encoding. Although the role of the classical network is for pre-processing, the majority of parameters that are learned through the training process is on the classical network side. One way of encoding all the entries of an image matrix would be iterating through smaller sections of the image, which naturally segues to convolutional operations. A combination of quantum convolutional network layers and quantum neural network layers is a way of implementing a purely quantum network.

In implementing machine learning algorithms from classical to quantum, the CV model offers many advantages over the qubit-based model. The quantum computational state space of a qumode in the CV model is infinite-dimensional while it is only 2-dimensional in the qubit-based model. The ability to define the dimension of the CV quantum computational space in approximating the original infinite space avails users of added freedom and flexibility in experimenting quantum algorithms. Photonic quantum computers, which are a version of physical implementation of the CV model, can be easily incorporated into the current

computing infrastructure because of their operability at room temperature.

Acknowledgements

The authors appreciate feedback and clarifying remarks from Maria Schuld and people at Xanadu.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Killoran, N., Bromley, T., Arrazola, J., Schuld, M., Quesada, N. and Lloyd, S. (2019) Continuous Variable Quantum Neural Networks, *Physical Review Research*, **1**, Article ID: 033063. <https://doi.org/10.1103/PhysRevResearch.1.033063>
- [2] Weedbrook, C., *et al.* (2012) Gaussian Quantum Information. *Reviews of Modern Physics*, **84**, 621. <https://doi.org/10.1103/RevModPhys.84.621>
- [3] Nielsen, M. and Chuang, I. (2010) Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.
- [4] Schuld, M., Bocharou, A., Svore, K. and Wiebe, N. (2018) Circuit-Centric Quantum Classifiers. *Physical Review A*, **101**, Article ID: 032308. <https://doi.org/10.1103/PhysRevA.101.032308>
- [5] Bergholm, V., *et al.* (2018) PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations. arXiv:1811.04968.
- [6] Braunstein, S. and Loock, P. (2005) Quantum Information with Continuous Variables. *Review of Modern Physics*, **77**, 513. <https://doi.org/10.1103/RevModPhys.77.513>
- [7] Knill, E., Laflamme, R. and Milburn, G.J. (2001) A Scheme for Efficient Quantum Computation with Linear Optics. *Nature*, **409**, 46-52. <https://doi.org/10.1038/35051009>
- [8] Neergaard-Nielsen, J. (2008) Generation of Single Photons and Schrodinger Kitten States of Light. PhD Thesis, Technical University of Denmark, Lyngby.
- [9] 18.06SC Linear Algebra Fall (2011) Singular Value Decomposition. MIT OpenCourseWare. <https://ocw.mit.edu/courses/18-06sc-linear-algebra-fall-2011/resources/lecture-29-singular-value-decomposition/>