

# Quantum Machine Learning: Variational Quantum Classifier

Noah Hamilton

December 6, 2023

## Contents

<b>1</b>	<b>What is Quantum Machine Learning?</b>	<b>2</b>
<b>2</b>	<b>Background: Classical Machine Learning</b>	<b>2</b>
<b>3</b>	<b>Parameterized Quantum Circuits</b>	<b>4</b>
<b>4</b>	<b>Data Encoding</b>	<b>6</b>
<b>5</b>	<b>Quantum Model</b>	<b>8</b>
<b>6</b>	<b>Optimization and Prediction</b>	<b>10</b>
<b>7</b>	<b>Simulation</b>	<b>11</b>
<b>8</b>	<b>Challenges and Future Work</b>	<b>13</b>

# 1 What is Quantum Machine Learning?

Quantum Machine Learning (QML) is a cross-section between Quantum Computing and Classical Machine Learning. Quantum Computing takes advantage of quantum mechanics to provide an advantage in time and cost over classical computers [9]. Machine Learning is a subset of the classical field known as Artificial Intelligence that focuses on using data and algorithms to imitate how humans learn [12]. This crossover between Quantum Computing and Machine Learning is being actively researched [9].

The interplay between Quantum and Classical is usually classified in two ways: what is the input data (Classical or Quantum) and what is the computing platform (Classical or Quantum). Decades of research in machine learning typically falls into the category of Classical Computing with Classical Data. Recent research efforts have been trying to take advantage of near-term quantum devices with classical data falling into the paradigm of Quantum Computing with Classical Data.

This research effort focuses specifically on a method in Quantum Machine Learning known as Variational Quantum Classifier (VQC). The main task of a VQC is to train a quantum circuit on labelled samples to predict labels for new data. A VQC embeds classical data into a quantum circuit, applies a parameterized model, produces a measurement of the circuit to extract labels, and updates the model using optimization techniques [1]. This research implements a VQC on three datasets and compares performance across different quantum models using Qiskit and scikit-learn.

To fully understand the context of a VQC, it is important to grasp the classical machine learning classification problem. This research will also use a known good classical machine learning technique known as Support Vector Machine (SVM) as a baseline for performance comparison.

# 2 Background: Classical Machine Learning

There are several types of machine learning: supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, and deep learning [2]. For the sake of brevity, this paper will focus on the concept of supervised learning, which requires the data to be pre-labeled so that the algorithm can see how accurate its performance is [2].

UC Berkeley School of Information describes supervised learning in three

steps [2]:

1. **A Decision Process:** A way to compute a prediction from a given input.
2. **An Error Function:** A way to check if the prediction is correct, and quantify how accurate it is.
3. **An Optimization Process:** A way to change the decision process to give better predictions.

The classification problem falls into this supervised learning paradigm where labelled data is used to train a classical model and used to predict new samples. The problem can take the form of binary classification or multi-class classification. A binary classification problem could be like distinguishing between cats and dogs, while a multi-class classification could be distinguishing between different types of flowers.

A well-known application of this problem is used for Optical Character Recognition, where a number/letter is hand-drawn and a computer attempts to guess which number/letter it is. More complicated applications of this classification problem can be used in RF Machine Learning which takes RF Data in the form of spectrograms (Power vs. Frequency) and attempts to predict the different types of modulation. All of these examples take lots of labelled input data, make a prediction, and then optimize based on the accuracy of the prediction. The formal definition of the classification problem can be found within the paper by Havlicek and others [10].

One of the more popular and successful algorithms is known as *Support Vector Machine* (SVM). The objective of SVM is use a linear decision boundaries to make predictions about the data. This solution is ideal for data that is linearly separable.

In Figure 1, an example SVM is used on the first two features in the Iris dataset [17] [16]. This research uses a very simple SVM for comparison and benchmarking in an image classification task.

Now that the stage is set from classical machine learning classification, it is time to dive into how a Variational Quantum Classifier works.

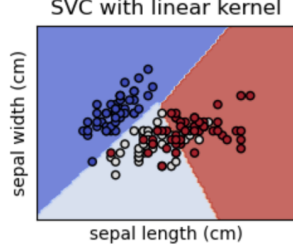


Figure 1: An Example SVM classifying data on the Iris dataset.

### 3 Parameterized Quantum Circuits

An essential part of many Quantum Machine Learning algorithms are Parameterized Quantum Circuits (PQCs). These are quantum Circuits with some fixed gates like controlled NOTs (CNOT) and adjustable gates like Qubit Rotations. These circuits are good candidates for demonstrating quantum advantage on near-term quantum devices because they can be optimized while taking into consideration gate and other environmental noise [3]. Figure 2 below is an example of a simple parameterized circuit that takes a single parameter  $\theta$  and then entangles  $q_0$  and  $q_1$

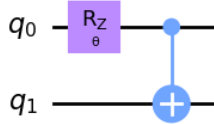


Figure 2: An Example Parameterized Quantum Circuit with Z Rotation with parameter  $\theta$  and a single CNOT gate. [13]

Since all gates used in a Quantum Circuit must be unitary, it is possible to describe a parameterized circuit as unitary operation on  $n$  qubits,  $U_\theta$ , acting on some initial state  $|\phi_0\rangle$ , often set to  $|0\rangle^{\otimes n}$  [13]. This leads to the resulting quantum state of the parameterized circuit:  $|\phi_\theta\rangle = U_\theta |\phi_0\rangle$  [13].

When using PQCs for machine learning algorithms, it is important that the circuit be as generalized as possible. Researchers recently proposed two measures to differentiate between different PQCs: expressibility and entangling capacity [18].

- *Expressibility*: The coverage of the Hilbert space by a circuit. Circuits with high expressibility can represent more unitaries. This can be quantified by the deviation from a uniform distribution [18] See Figure 3
- *Entangling Capacity*: Entanglement is a key resource in quantum computing. Meyer-Wallach defined a metric for how entangled a state is, with 0 being unentangled and 1 being maximally entangled (i.e. a Bell State) [4]. Sim and others defined Entangling Capacity as the average Meyer-Wallach measure. See Figure 4

It has been shown that there is a strong correlation between expressibility and classification accuracy when using a Variational Quantum Classifier [11]. Another consideration for selecting a PQC for a machine learning model is hardware efficiency. To have any real impact in the near-term quantum computing era, the circuit depth is very important and directly impacts the accuracy of many classifiers.

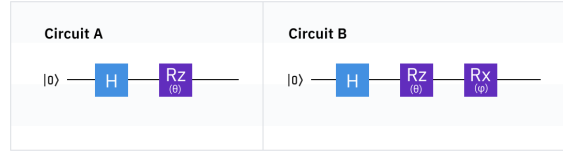


Figure 3: Example PQCs with different expressibility. Circuit A is low, while Circuit B is high. This means that Circuit A can explore fewer states than Circuit B. [13]

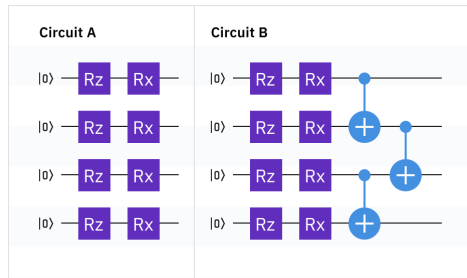


Figure 4: Example PQCs with different entangling capacity. Circuit A is low, while Circuit B is high. This means that Circuit A is more limited in generating certain entangled states. [13]

PQCs are used mainly for two things in QML: *encoding data* (where parameters are the input data) and as a *Quantum Model* where parameters are optimized in the training process.

## 4 Data Encoding

In a Variational Quantum Classifier, it is important to have a way to map input data into a quantum circuit. This is typically done with a PQC. Data Encoding is also referred to as data embedding, quantum embedding, or quantum reference state, where classical data is mapped into a quantum circuit using a quantum feature map. The goal is to take classical data  $x$  and map into  $|\psi_x\rangle$  using a set of gate parameters [15]. Some simple data embeddings are shown below. The *ZZ Feature Map* is what used in this implementation of a VQC, shown later.

### Basis Embedding

Basis embedding maps each input with a computational basis state of a qubit system, hence the classical data must be able to be represented in binary strings. An entire dataset could be represented by the following equation assuming  $D$  is a dataset with  $x_n$ : [8]

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^M |x^{(m)}\rangle$$

Basis embedding is a good option for discrete data represented as binary or integers [14].

### Amplitude Embedding

Amplitude Encoding is similar to basis encoding but instead of each input receiving its own computational basis, data is encoded into the amplitudes of a quantum state. It requires the classical dataset to be normalized so that each quantum state is valid. This embedding can represent a normalized classical  $N$ -dimensional data point  $x$  by the amplitudes of a  $n$ -qubit state  $|\psi_x\rangle$  as: [8]

$$|\psi_i\rangle = \sum_{i=1}^{2^n} x_i |i\rangle$$

Amplitude embedding is superb for modelling complex continuous data represented by complex values [14].

## Angle Embedding

A simple encoding is also the angle embedding because it is simply a rotation along one of the axis of Rotation of the Bloch sphere. The following mapping demonstrates how encode data into a single qubit:

$$x \mapsto R_k(x)|0\rangle = e^{-ix\sigma_k/2}|0\rangle,$$

Angle embedding is great for continuous floating point data; however angle embedding typically leads to higher cost for optimization and construction [14].

## ZZ Feature Map

Each of the above techniques can be combined to create more complicated and higher order embeddings. These embeddings can also come with Hadamard gates interleaved with entangling blocks. Researchers have found that a specific PQC known as *ZZFeatureMap* is very effective on near-term hardware [10]. Below in Figure 5, two data points are encoded into a quantum circuit. To expand to more data points, simply perform the Pauli Rotation with the next qubit and entangle with other qubits. Figure 6 shows the expansion of this circuit to 3 qubits. This is the circuit that will be used later in the simulation of the VQC.

Now that the input data has been encoded into a quantum circuit using one of the above techniques, it is time to implement a quantum model.

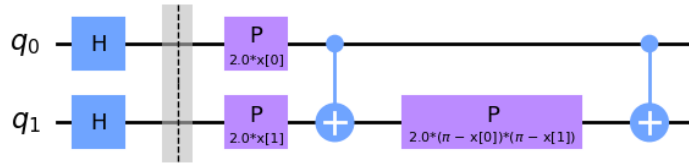


Figure 5: Example of 2 data points  $x_0$  and  $x_1$  encoded into a ZZFeatureMap circuit using Qiskit

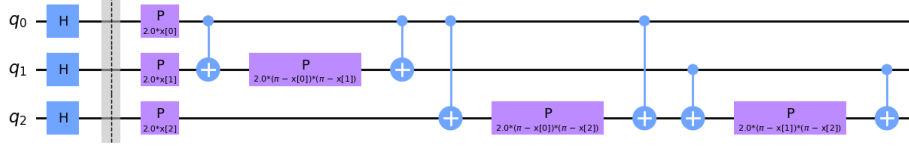


Figure 6: Example of 3 data points  $x_0$ ,  $x_1$ , and  $x_2$  encoded into a ZZFeatureMap circuit using Qiskit

## 5 Quantum Model

A quantum model is simply a specific parameterized circuit that can be optimized to increase accuracy. In the above section 3, these circuits are explained in detail. In this section, Heuristics, trade-offs and examples will be shown.

IBM researchers outline three trade-offs to consider when choosing a specific quantum model, and of course there is always a trade-off between speed and quality. More parameters within the circuit is likely to become more accurate, but will take longer to run [7].

1. **Speed:** Limiting the search space decreases the time it takes to run the algorithm (recall previous discussion of "expressibility")
2. **Accuracy:** Limiting the search space risks excluding the correct solution, leading to suboptimal results.
3. **Noise:** Deep Circuits are impacted by noise heavily.

Let's dive into some examples including N-Local Circuits, Two-Local, Real Amplitudes, and Efficient SU(2). For each example, the quantum circuit will be shown along with a brief description.

### N-Local Circuits

N-Local Circuits (Figure 7) are the most used heuristic models used and all the following circuits are derived from N-Local Circuits. These circuits are great for quantum models because they have the best of all worlds when it comes to an efficient implementation while capturing the most important correlations. The N represents the number of layers of rotation gates [7].



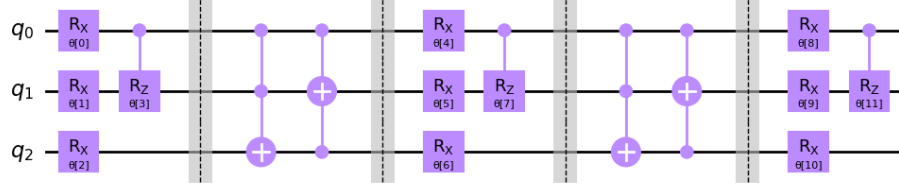


Figure 7: Example of 3 Qubit NLocal circuit using Qiskit

## Two-Local Circuits

Two Local Circuits (Figure 8) are specific instance of N-Local Circuits where  $N$  is set to 2. This circuit consists of a single-qubit rotations and 2-qubit entanglement gates. The types of rotations can be interchanged between  $R_x$ ,  $R_y$  and  $R_z$ , as well as the number of repetitions of this circuit [7].

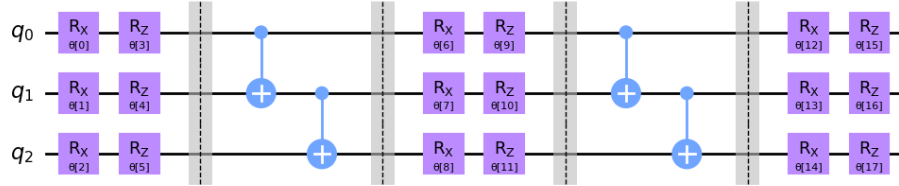


Figure 8: Example of 3 qubit TwoLocal circuit using Qiskit

## Real Amplitudes

Real Amplitudes (Figure 9) is a specific instance of a Two-Local Circuit that has been used in chemistry applications alongside classification problems. Since all the amplitudes are real, there is no complex parts. The circuit specifically uses  $Y$  rotations and  $CX$  entanglements [7].

## Efficient $SU(2)$

Efficient  $SU(2)$  (Figure 10) is also a specific version of a Two-Local circuit that uses  $SU(2)$  and  $CX$  entanglement, where  $SU(2)$  is a special unitary group of degree 2 such as a Pauli Rotation. This is a well known quantum model for machine learning on near-term devices because it has high expressibility and is very efficient on hardware [7].

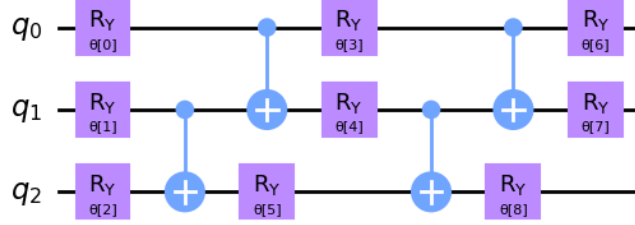


Figure 9: Example of 3 qubit Real Amplitudes Circuit using Qiskit

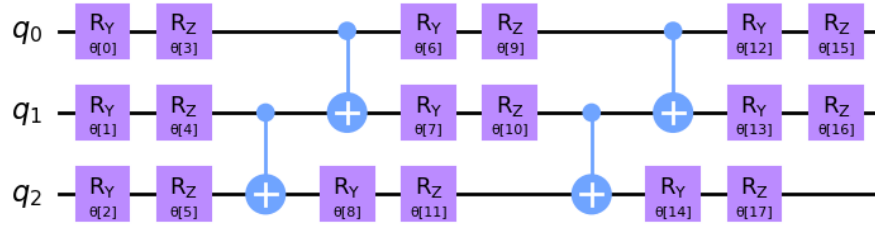


Figure 10: Example of 3 qubit EfficientSU2 circuit with 2 repetitions using Qiskit

The last three of these quantum models were used as the quantum model within the simulated VQC shown later.

## 6 Optimization and Prediction

This research isn't focused on the optimization and training process of these circuits. Briefly, it is notable that classical optimization algorithms can be used to tune the parameterized quantum circuits. Techniques like gradient descent get generalized into a more efficient version using the Parameter Shift Rule [5].

Prediction and label assignment comes from probabilistic simulation of many shots of these circuits. Specifically, the predicted class comes as the computational state with the highest probability. It is possible to define other functions to predict a class, but all of these methods rely on performing a measurement in the computational basis.

## 7 Simulation

Now, let's get into the fun part of investigating these Variational Quantum Classifiers.

### Starting Point and Setup

This research stems off of educational material from Qiskit [6]. The major changes from the initial tutorial code include the expansion to other datasets aside from the Iris Dataset as well as experimentation with different Ansatz Functions. The libraries used in this Python simulation are Qiskit and sklearn. Qiskit is meant for quantum programming while sklearn implements many classical machine learning functions.

### Datasets

Three datasets were used for performance evaluation [16]. These are classic machine learning datasets: Iris, Wine, and Digits. For this effort, condensed versions were used to decrease computation time. Important notes about the datasets are found below:

- **Iris:** 150 samples, 4 Features, 3 Classes, Numeric Data about sepal length, sepal width, petal length, and petal width.
- **Wine:** 178 samples, 13 Features, 3 Classes, Numeric Data about different wine attributes.
- **Digits:** 1797 samples, 17 Features, 10 Classes, Images of Handwritten digits 8×8 values of 0-16 integers.

### Implementation

This section will detail the steps taken to implement the Variational Quantum Classifier. The full code will be attached with submission.

1. Load the Data using 'sklearn' functions.
2. Train a Classical Machine Learning Model. In this research, a basic SVM was used with default parameters for comparison.
3. Train a Quantum Machine Learning Model.
  - (a) Determine number of features from input data

- (b) Select a feature map. In this case, ZZFeatureMap was used (discussed above 4)
- (c) Select a quantum model or Ansatz. 3 quantum models were compared (TwoLocal, RealAmplitudes, and EfficientSU2)
- (d) Select optimization and simulator functions. (This was not the focus off the research, so a default was chosen ‘COBYLA’ and the Qiskit ‘Sampler’ primitive )
- (e) Train the model using ‘sklearn’ “fit()” function
- (f) Evaluate the model using ‘sklearn’ “score()” function

## Results

Above is the basic implementation steps that were taken in conducting this computer simulation. The following tables 11 show the results on each of the datasets. One thing that was examined was the use a classical machine learning technique for dimensionality reduction known as Principal Component Analysis (PCA). This reduced the input features to the VQC significantly and allowed the simulation to run much quicker and sometimes increased the accuracy.

It can be seen that the TwoLocal Ansatz implementation seemed to have good results on the Wine and Iris datasets. The Digits dataset seemed to have poor performance overall and part of that is because of how the image dataset is encoded into the circuit.

Results on the IRIS Dataset		
Model	Test Score	Train Score
SVC, 4 features	0.99	0.97
VQC, 4 features, RealAmplitudes	0.85	0.87
-----		
SVC, 2 features	0.97	0.90
VQC, 2 features, RealAmplitudes	0.58	0.63
VQC, 2 features, EfficientSU2	0.71	0.63
VQC, 2 features, TwoLocal	0.76	0.73
Results on the WINE Dataset		
Model	Test Score	Train Score
SVC, 12 features	1.00	0.97
VQC, 12 features, RealAmplitudes	0.59	0.44
-----		
SVC, 2 features	0.99	0.97
VQC, 2 features, RealAmplitudes	0.48	0.42
VQC, 2 features, EfficientSU2	0.73	0.78
VQC, 2 features, TwoLocal	0.81	0.64
Results on the DIGITS Dataset		
Model	Test Score	Train Score
SVC, 64 features	1.00	0.99
VQC, 64 features, RealAmplitudes	ERROR: Too Many Features   ERROR: Too Many Features	
-----		
SVC, 2 features	0.67	0.69
VQC, 2 features, RealAmplitudes	0.19	0.16
VQC, 2 features, EfficientSU2	0.20	0.19
VQC, 2 features, TwoLocal	0.11	0.12

Figure 11: Screenshot of results for each of the three datasets

## 8 Challenges and Future Work

Overall, the performance of this Variational Quantum Classifier is not superior to classical machine learning, and one of the main reasons is lack of training time. If more training cycles were used and more data was available, the accuracy would be improved. Another challenge is that the 64 features that were inputted when the ‘digits’ dataset was used caused the VQC to not compile using the IBM Sampler. Future work would include investigation in better optimization algorithm that are “*Quantum Aware*”.

In conclusion, Variational Quantum Classifiers are an exciting field of research in Quantum Machine Learning as they can be implemented on near-term quantum computers to provide valuable insights in the world today.

## References

- [1] Amira Abbas. Build a quantum classifier. IBM Qiskit Summer School, 2021.
- [2] Data Science at UC Berkeley. What is machine learning (ml)? data-science@berkeley, the online Master of Information and Data Science from UC Berkeley, 2022.
- [3] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, November 2019.
- [4] Gavin K. Brennen. An observable measure of entanglement for pure states of multi-qubit systems, 2003.
- [5] Gavin E. Crooks. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition, 2019.
- [6] IBM Qiskit Developers. Training a quantum model on a real dataset. Qiskit Ecosystem, 2023.
- [7] Qiskit Developers. Ansatzes and variational forms. IBM Quantum Learning, 2023.
- [8] Xanadu Developers. Quantum embeddings, 2023.
- [9] David Peral García, Juan Cruz-Benito, and Francisco José García-Peñalvo. Systematic literature review: Quantum machine learning and its applications, 2022.
- [10] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, March 2019.
- [11] Thomas Hubregtsen, Josef Pichlmeier, Patrick Stecher, and Koen Bertels. Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility and entangling capability, 2020.
- [12] IBM. What is machine learning?
- [13] qiskits developers. Parameterized quantum circuits, 2023.

- [14] Isdor Schoch (Xanadu Resident). How to embed data into a quantum state. PennyLane Blog, 2022.
- [15] Maria Petruccione Schuld. *Supervised learning with Quantum Computers*. Springer, 2019.
- [16] scikit-learn developers. 7.1 toy datasets, 2007-2023.
- [17] scikit-learn developers. Plot different svm classifiers in the iris dataset, 2007-2023.
- [18] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12), October 2019.