Quantum Variational Classifier

Ananth Krishna Kidambi
CSE department
Indian Institute of Technology, Bombay
Mumbai, India

***Abstract*—This paper presents a study of the application of quantum machine learning techniques using variational classifiers. By leveraging the advantages of quantum computing, such as the ability to represent and manipulate high-dimensional data, we demonstrate significant improvements in the performance of variational classifiers for a range of tasks. Our results show that the use of quantum computing can lead to faster and more accurate classification, making it a promising approach for a wide range of applications in machine learning and artificial intelligence.**

***Index Terms*—Quantum computing, machine learning, artificial intelligence.**

I. Introduction

Quantum computing is a field of computing that utilizes the principles of quantum mechanics to perform computations. Unlike classical computers, which store and process information using bits (binary digits) that can be in one of two states (0 or 1), quantum computers use quantum bits, or qubits, which can exist in a superposition of states. This allows quantum computers to perform certain types of calculations much faster and more efficiently than classical computers.

Machine learning is a subfield of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed. The field has its roots in the 1950s and 60s, but has seen a resurgence in recent years due to advances in computer hardware and the availability of large amounts of data. Supervised learning involves training a model on a labeled dataset, where the desired output is already known. This type of learning is used for tasks such as image classification and spam detection.

Quantum machine learning is a relatively new field that combines the principles of quantum computing and machine learning. It aims to leverage the unique properties of quantum systems, such as superposition and entanglement, to perform machine learning tasks more efficiently and effectively than classical methods.

One of the main goals of quantum machine learning is to find ways to speed up the training and inference of machine learning models using quantum computers. This can be achieved by using quantum algorithms to perform tasks such as linear algebra, optimization, and feature extraction, which are commonly used in machine learning.

Quantum algorithms, which can be implemented in time polynomial in the number of qubits (n), can perform computations on 2^n amplitudes. This can result in polylogarithmic running times. However, most algorithms using amplitude encoded quantum machine learning models require non-trivial quantum subroutines which cannot be implemented on small-scale devices.

Hybrid approaches called ”variational algorithms are much more suited to near term quantum computing, since they require a fewer number of qubits.

In this paper, we implement a quantum variational classifier, a supervised learning model, and train it to classify iris images. We use amplitude encoding to encode real vectors into qubits, before processing them.

II. Problem Statement

Let X be set of inputs and Y be a set of outputs. Given a dataset $D = \{(x^1, y^1), ..., (x^M, y^M)\}$ of pairs of *training inputs* $x^m \in X$ and *target outputs* $y^m \in Y$ for $m = 1, ..., M$, our goal is to predict a new output y for a new input x.

For the sake of simplicity, we will assume that $X = \mathbb{R}^N$ and $Y = \{0, 1\}$, that is, a binary classification on a N-dimensional input space.

III. The Circuit-Centric Classifier

The most common way of solving this problem classically is to first train a model $f(x, \theta)$ by adjusting a set of parameters θ, and then use the trained model to predict y.

The key idea here is to use a quantum circuit consisting of single and 2-qubit gates in the model that we use for classification. In order to achieve this, there are four major steps-

1) First, we encode the classical data into the states of the qubits. For this, we use amplitude encoding.
2) Then, we apply the quantum model to the initial state. This can be represented using a unitary transformation ($|\varphi'\rangle = U_\theta |\varphi(x)\rangle$).
3) Third, the prediction is read from the final state $|\varphi\rangle$. Using repeated measurements, we can find the probability of getting 1.
4) Lastly, we add a bias parameter b and map the result to the output $y \in \{0, 1\}$.

More formally, we are defining a classifier that takes decisions according to

$$f(x; \theta, b) = \begin{cases} 1 & \text{if } \sum_{i=2^{n-1}+1}^{2^n} |(U_\theta \varphi(x))_i|^2 + b > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where, φ is the state obtained in the first step.

A. State Preparation

In the most general terms, state preparation implements a feature map $\varphi : mathbbR^N \rightarrow \mathbb{C}^{2^N}$, where N is the total number of qubits used to represent the features. In the following we focus on *amplitude encoding*, where an input vector $x \in \mathbb{R}^N$, possibly with some further preprocessing to bring it into a suitable form, is directly associated with the amplitudes of the 2^N-dimensional 'ket' vector of the quantum system written in the computational basis.

For this map to be valid, we require that the output vector is normalized ($x^T x = 1$). This can pose a challenge, since the proximity relations between vectors may depend on the length of the vectors as well as the angles, and hence some data sets can be distorted by normalization. One possible solution is to embed the vector in a higher dimensional space. This can be achieved by adding non-zero padding terms before normalization.

B. The Model

The model maps the encoded feature vector $\phi(x)$, which is now a ket vector in the Hilbert space, to another ket vector $|\varphi'\rangle = U_\theta |\varphi(x)\rangle$ by a unitary operation U_θ, which is parametrised by a set of variables θ.

As described earlier, we express U as a product of single qubit operations and 2-qubit quantum gates. We can represent a single qubit gate G_k as

$$U_L = \mathbb{I}_0 \otimes \cdots \otimes G_k \otimes \cdots \otimes \mathbb{I}_{N-1}$$

It is known that single-qubit gates, together with any set of imprimitive 2-qubit gates provide for quantum universality:

Observation 1. *Circuits of the form $U = U_L \dots U_k \dots U_1$ composed out of single-qubit gates and at least one type of imprimitive 2-qubit gates generate the entire unitary group $U(2^N)$ in a topological sense. That is, for any $\epsilon > 0$ and any unitary $V \in U(2^N)$ there is a circuit of the the form descibed above, the value of which is ϵ-close to V.*

In order to add parameters to the circuit that can be trained, we express the single qubit gates G (which are 2×2 unitary matrices) as

$$G(\alpha, \beta, \gamma, \delta) = e^{i\phi} \begin{pmatrix} e^{i\beta}\cos(\alpha) & e^{i\gamma}\sin(\alpha) \\ e^{i\gamma}\sin(\alpha) & e^{i\beta}\cos(\alpha) \end{pmatrix} \tag{2}$$

which is fully defined by 4 parameters $\{\alpha, \beta, \gamma, \delta\}$. Also, since global phases cannot be measured, we can neglect the prefactor $e^{i\phi}$, and hence have only 3 trainable parameters.

C. Measurement and postprocessing

The measurement of the first qubit results in the state 1 with the following probability-

$$p(q_0 = 1; x, \theta) = \sum_{i=2^{n-1}+1}^{2^n} |(U_\theta \varphi(x))_i|^2 \tag{3}$$

In order to estimate this probability, we run the circuit multiple times and then calculate $p(q_0 = 1; x, \theta)$ from the result.

Next, we add a learnable bias term b to produce the continuous output of the model,

$$\pi(x; \theta, b) = p(q_0 = 1; x, \theta) + b \tag{4}$$

Thresholding the value finally yields the binary output that is the overall prediction of the model:

$$f(x; \theta) = \begin{cases} 1 & \text{if } \pi(x; \theta, b) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

In Dirac notation, this can be written as

$$\pi(x; \theta, b) = \frac{1}{2}(E(\sigma_z) + 1) + b \tag{6}$$

where

$$E(\sigma_z) = \langle\varphi'|\, \sigma_z \,|\varphi'\rangle$$

IV. Circuit Architecture

Since we need to build a low depth cicuit with single qubit or controlled single qubit gates, a natural approach is to consider circuits that are strongly entangling, because such circuits can reach 'wide corners of the Hilbert space' with $U_\theta |0 \dots 0\rangle$. Reversibly argued, they have a better chance to project input data state $|\varphi(x)\rangle$ with the class label y onto the subspace $|y\rangle \otimes |\eta\rangle, \eta \in \mathbb{C}^{2^{N-1}}$, which corresponds to a decision of $p(q_0) = 0, 1$ in our classifier (for a zero bias). Also, such a circuit can capture both the short-range corelations and the long-range corelations in the input data.

We compose the circuit out of several blocks. Each block consists of a single qubit gate $G(\alpha, \beta, \gamma)$ applied on each qubit, followed by a layer of controlled gates. For $j \in \{1 \dots \frac{N}{gcd(N,r)}\}$, where r is the range of the control, the jth qubit of a block has the ($(jr - r)$mod$N$)th qubit as its target and the (jrmod$N$)th qubit as its control. Assuming that r in co-prime with N, we can entangle/unentangle all qubits using this circuit. Hence, a data block has the form

$$B = \prod_{k=0}^{N-1} C_{C_k}(G_{t_k}) \prod_{j=0}^{N-1} G_j \tag{7}$$

We can optimize this circuit further using the following observation-

Observation 2. *A circuit block of the form (7) can, up to global phase, be uniquely rewritten as*

$$B = \prod_{k=0}^{N-1} R_k^X C_{C_k}(P_{t_k}) \prod_{j=0}^{N-1} G_j \tag{8}$$

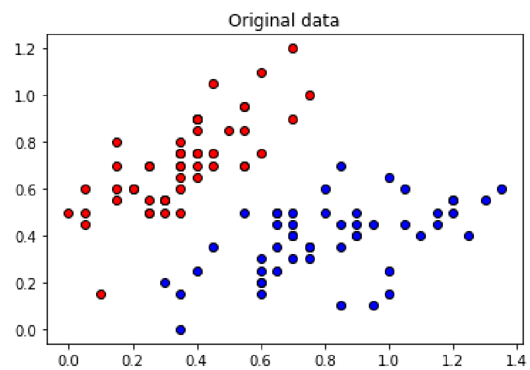
where are single qubit gates with the usual three parameters(and, moreover, G_j is an axial rotation for $j > 0$), P is a single-parameter phase gate, and R^X is a single-parameter X-rotation.

V. Training

In order to train the model, we use gradient descent. However, since we are evaluating the model function on a quantum device, we don't have any classical access to the derivatives. Hence, we need a hybrid scheme that combines classical updates of the parameters, and quantum information processing to extract the gradients.

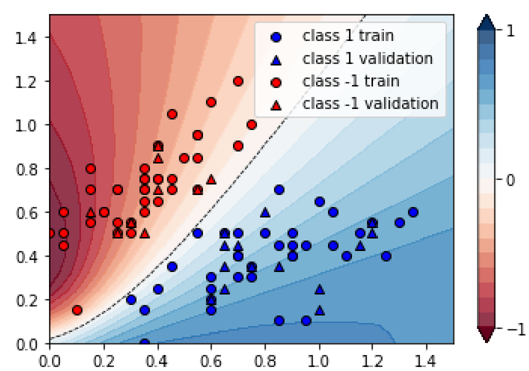
VI. Results

The following data from the iris dataset was used to train and test the model.



A model with 6 layers and 2 qubits was used as the variational classifier.

The following result was obtained after training the model and testing:



References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] https://arxiv.org/abs/1804.00633v1