Quantum Circuit Learning (QCL)

Andrey Yashkin

Novosibirsk State University

2020

Outline

Terminology

Density operator

The density operator language provides a convenient means for describing quantum systems whose state is not completely known. More precisely, suppose a quantum system is in one of a number of states $|\phi_i\rangle$, where i is an index, with respective probabilities p_i. We shall call $\{p_i, |\phi_i\rangle\}$ an ensemble of pure states. The density operator for the system is defined by the equation

$$\rho \equiv \sum_i p_i |\phi_i\rangle\langle\phi_i|$$

Quantum system evolution

The evolution of a closed quantum system is described by the unitary operator U. If the system was initially in the state $|\phi_i\rangle$ with probability p_i then after the evolution has occurred the system will be in the state $U|\phi_i\rangle$ with probability p_i. Thus, the evolution of the density operator is described by the equation

$$\rho \equiv \sum_i p_i |\phi_i\rangle\langle\phi_i| \xrightarrow{U} \sum_i p_i U|\phi_i\rangle\langle\phi_i|U^\dagger \equiv U\rho U^\dagger$$

Expectation value

In quantum mechanics, the expectation value is the probabilistic expected value of the result (measurement) of an experiment. Consider an operator A. The expectation value is then $\langle A \rangle_\psi = \langle \psi | A | \psi \rangle$ in Dirac notation with $|\psi\rangle$ a normalized state vector of the quantum system state.
A similar formula holds for the density operator ρ

$$\langle A \rangle_\rho = Tr(\rho A) = \sum_i p_i \langle \psi_i | A | \psi_i \rangle = \sum_i p_i \langle A \rangle_{\psi_i}$$

Composite systems

The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state $|\phi_i\rangle$ then the joint state of the total system is $\bigotimes_{k=1}^{n} |\phi_i\rangle$

Algorithm

Input data

QCL framework aims to perform supervised or unsupervised learning tasks. In supervised learning, an algorithm is provided with a set of input x_i and corresponding teacher data $f(x_i)$. The algorithm learns to output $y_i = y(x_i, \theta)$ that is close to the teacher $f(xi)$, by tuning θ. The output and the teacher can be vector-valued. The objective of learning is to minimize a cost function, which is a measure of how close the teacher and the output is, by tuning θ.

QCL algorithm on N qubit circuit

I Encode input data x_i into some quantum state $|\phi_i n(x_i)\rangle$ by applying a unitary input gate $U(x_i)$ to initialized qubits $|0\rangle$

II Apply a θ-parameterized unitary $U(\theta)$ to the input state and generate an output state $|\phi_{out}(x_i, \theta)\rangle = U(\theta)|\phi_{in}(x_i)\rangle$.

III Measure the expectation values of some chosen observables. Specifically, we use a subset of Pauli operators $\{B_j\} \subset \{I, X, Y, Z\}^{\otimes N}$. Using some output function F, output $y_i = y(x_i, \theta)$ is defined to be $y(x_i, \theta) \equiv F(\{B_j(x_i, \theta)\})$.

IV Minimize the cost function $L(f(x_i), y(x_i, \theta))$ of the teacher $f(x_i)$ and the output y_i , by tuning the circuit parameters θ iteratively.

V Evaluate the performance by checking the cost function with respect to a data set that is taken independently from the training one.

Ability to approximate a function

Input data are one dimension

Let x and $\rho_{in}(x) = |\psi_{in}(x)\rangle\langle\psi_{in}(x)|$ be an input data and a corresponding density operator of input state. $\rho_{in}(x)$ can be expanded by a set of Pauli operators $\{P_j\} = \{I, X, Y, Z\}^{\bigotimes N}$ with $a_k(x)$ as coefficients, $\rho_{in}(x) = \sum_k a_k P_k$. A parameterized unitary transformation $U(\theta)$ acting on $\rho_{in}(x)$ creates the output state, which can also be expanded by $\{Pk\}$ with $b_k(x)$. Now let $u_{ij}(\theta)$ be such that $b_m(x, \theta) = \sum_k u_{mk}(\theta) a_k(x)$. b_m is an expectation value of a Pauli observable itself.

Input data are one dimension

When the teacher $f(x)$ is an analytical function, we can show, at least in principle, QCL is able to approximate it by considering a simple case with an input state created by single-qubit rotations. The tensor product structure of quantum system plays an important role in this analysis. Let us consider a state of N qubits: $\rho_{in}(x) = \frac{1}{2^N} \bigotimes_{i=1}^{N} [I + xX_i + \sqrt{1-x^2} Z_i]$ This state can be generated for any $x \in [-1, 1]$ with single-qubit rotations, namely, $\prod_{i=1}^{N} RY(sin^{-1} x)$

Observables

The state given before has higher order terms up to the Nth with respect to x. Thus an arbitrary unitary transformation on this state can provide us with an arbitrary N th order polynomial as expectation values of an observable. Terms like $x\sqrt{1-x^2}$ in the state can enhance its ability to approximate a function.
Important notice in the example given above is that the highest order term x^N is hidden in an observable $X^{\otimes N}$. To extract x^N from the state, one needs to transfer the nonlocal observable $X^{\otimes N}$ to a single-qubit observable using entangling gate such as the controlled-NOT gate. Entangling nonlocal operations are the key ingredients of the nonlinearity of an output.

$$\langle X^{\otimes N} \rangle_{\rho_{in}(x)} = Tr(\rho_{in}(x) X^{\otimes N}) = x^N$$

Multi-dimensional inputs

The above argument can readily be generalized to multi-dimensional inputs. Assume that we are given with d-dimensional data $x = x_1, x_2, \ldots, x_d$ and want higher terms up to the n_kth $(k = 1, \ldots, d)$ for each data, then encode this data into a $N = \sum_k n_k$ -qubit quantum state as $\rho_{in}(x) = \frac{1}{2^N} \bigotimes_{k=1}^{N} (\bigotimes_{i=1}^{n_k} [I + xX_i + \sqrt{1 - x^2} Z_i])$ These input states automatically has an exponentially large number of independent functions as coefficient set to the number of qubits. The tensor product structure of quantum system readily "calculates" the product such as $x_1 x_2$.

Optimization procedure

Gradiente calculation

To calculate a gradient of an expectation value of an observable with respect to a circuit parameter θ, suppose the unitary $U(\theta)$ consists of a chain of unitary transformations $\prod_{j=1} U_j(\theta_j)$ on a state ρ_{in} and we measure an observable B. For convenience, we use notation $U_{j:k} = U_j \dots U_k$. Then $\langle B(\theta) \rangle$ is given as $\langle B(\theta) \rangle = Tr(BU_{l:1}\rho_{in}U_{l:1})$. We assume U_j generated by a Pauli product P_j, that is, $U_j(\theta_j) = exp(-i\theta_j P_j/2)$. The gradient is calculated to be

$$\begin{aligned} \frac{\partial \langle B(\theta) \rangle}{\partial \theta_j} &= \frac{1}{2} Tr[BU_{l:j+1} U_j(\tfrac{\pi}{2}) \rho_j U_j^\dagger(\tfrac{\pi}{2}) U_{l:j+1}^\dagger] - \\ &\frac{1}{2} Tr[BU_{l:j+1} U_j(-\tfrac{\pi}{2}) \rho_j U_j^\dagger(\tfrac{-\pi}{2}) U_{l:j+1}^\dagger] \end{aligned} \tag{1}$$

where $\rho_j = U_{j:1}\rho_{in}U_{j:1}^\dagger$

Numerical simulations

Classification proble

As a demonstration, the classification problem, which is an important family of tasks in machine learning, is performed.

Figure

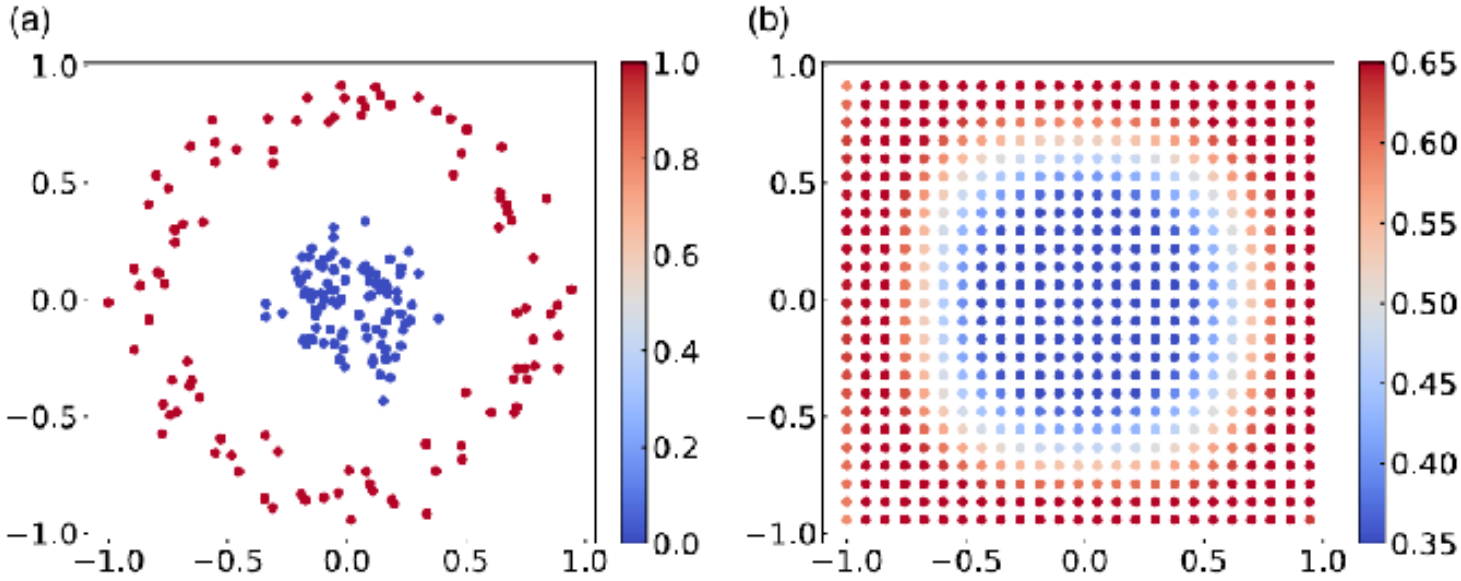

Figure: Demonstration of a simple nonlinear classification task. (a) teacher data. Data points that belong to class 0, 1 is shown as blue and red dot, respectively. (b) Optimized output from first qubit (after softmax transformation). 0.5 is the threshold for classification, less than and greater than 0.5 means that the point is classified as class 0 and 1, respectively.

Results

The figure shows the training data set, blue and red points indicate class 0 and 1 respectively. Here we train the quantum circuit to classify based on each training input data points $x_i = (x_{i,0}, x_{i,1})$. We define the teacher $f(x_i)$ for each input x_i to be two dimensional vector $(1, 0)$ for class 0, and $(0, 1)$ for class 1. The number of teacher samples is 200 (100 for class 0, and 100 for class 1). We see that QCL works as well for the nonlinear classification tasks.

Approach

The output is taken from the expectation value of the Pauli Z operator of the first 2 qubits, and they are transformed by softmax function F. For d-dimensional vector q, softmax function returns d-dimensional vector $F(q)$ with its kth element being $F_k(q) = e^{q_k} / \sum_i e^{q_i}$. Thus the output $y_i = (y_{i,0}, y_{i,1})$ is defined by $y_i = F(\langle Z_1(x_i, \theta_i) \rangle, \langle Z_2(x_i, \theta_i) \rangle)$

Conclusion

Conclusion

A machine learning framework on near-term realizable quantum computers was presented. Given method fully employs the exponentially large space of the quantum system, in a way that it mixes simply injected nonlinear functions with a low-depth circuit to approximate a complex nonlinear function. Numerical results have shown the ability to represent a function, to classify, and to fit a relatively large quantum system. Also, the theoretical investigation has shown QCL's ability to provide us a means to deal with high dimensional regression or classification tasks, which has been unpractical on classical computers.