RKHS endowed with Quantum Kernel

1. Building RKHS

Quantum kernel has the form
$$k(x, x') =| \langle 0|\phi^{\dagger}(x')\phi(x) |0\rangle |^2= \langle \Phi(x), \Phi(x')\rangle_{HS} = tr(\Phi(x)\Phi(x')).$$

To prove that a space spanned by the quantum kernel is RKHS, all we have to do is just to find that the Quantum kernel is positive definite kernel.

> A quantum kernel $k(x, x')$ made by any arbitrary quantum circuit is positive definite kernel.

Positive definiteness i.e. $\sum \sum v_i \bar{v}_j k(x_i, x_j) > 0$ for any x_i, x_j where $(x_1, x_2, \ldots x_n) \in \mathcal{X}^{\otimes n}$ and v_i, v_j where $(v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$

pf)

$$\sum_i \sum_j v_i \bar{v}_j \langle \Phi(x_i), \Phi(x_j)\rangle_{HS} = \sum_i \sum_j \langle v_i \Phi(x_i), v_j \Phi(x_j)\rangle = \langle \sum_i v_i \Phi(x_i), \sum_j v_j \Phi(x_j)\rangle$$

Because pairs of x_i, x_j and v_i, v_j come out from same pool, we can see that
$\sum_i v_i \Phi(x_i) = \sum_j v_j \Phi(x_j)$.

So if $\langle \Phi(x), \Phi(x)\rangle_{HS} > 0$ for any x.

This is obvious because $tr(\Phi(x)\Phi^{\dagger}(x)) = tr(I) = 2^n > 0$

The reproducing property of RKHS i.e. for $f(x) = \sum a_n k(x, x_n)$, $\langle f, k(\cdot, x_i)\rangle_H = f(x_i)$ holds naturally by defining the inner product as $\langle k(\cdot, x), k(\cdot, x')\rangle_H = k(x, x')$. The positive definiteness property of the kernel makes the kernel working as inner product.

The characteristics of quantum RKHS

- Customized Quantum Kernel
    - There is no restriction on the shapes of the quantum gate $\phi(x)$, we can build various types of RKHS with various quantum kernels. Therefore, the quantum kernel RKHS exists infinitely many.
    - This can lead to the customized RKHS according to types of data. Remark that there is only one custom setting γ at the Gaussian kernel ($exp(-\frac{1}{\gamma} \parallel x - x' \parallel_2^2)$)
- Directly calculating inner product
    - The classical kernel depends on the mercer's theorem which states that the mercer's kernel could be decomposed with the product of eigenfunction $k(x, x') = \sum_{i=1}^{\infty} \gamma_i \phi_i(x)\phi_i(y)$. Therefore, the advantage arises that the mercer's kernel is equivalent with the product sum of infinitely many function $\phi_i(x)$(Let's call it component function). That is, it can compute huge task with simpler function.
    - However, if we use quantum kernel, it directly defines the function $\phi_i(x)$ and calculate it with quantum circuit. Because the kernel and matrix multiplication is natural process of the quantum device, it can benefits over the classical kernel calculation (**Might be able to depict it analytically**)

- One of the potential topic about the kernel method is in using finitely many specific component function. Even though the mercer's theorem states that it use infinitely many function, in actually it only use just few of the component function because γ_i banishing as $i \to \infty$. The obstacles of it is huge calculation. If we can depict such specific component function in $U(x)\left|0\right\rangle$(: vector valued function), then we can use quantum supremacy about matrix multiplication on the problem. See that $U(x)\left|0\right\rangle$ is $2^n \times 1$ vector. This is not infinitely many but finitely many space enough to build specific component functions. (김일문 교수님의 조언).

Universality Analysis

Universality of VQC

According to [Schuld M. (2021) The effect of data encoding on the expressive power], VQC model $\hat{f}(x) = \left\langle 0 | U^\dagger(X) M U(X) \right| 0 \rangle$ could be rewritten as Fourier Series $\hat{f}(x) = \sum\limits_{j,k \in D} \alpha_{j,k} e^{i(\Lambda_j - \Lambda_k)x}$.

$\Lambda_j - \Lambda_k$ is determined by encoding gate e^{iHx} in $U(x)$ and $\alpha_{j,k}$ is determined by observable M and additional gate of $U(x)$.

If the term $\Lambda_j - \Lambda_k$ can make sequence of integer $\{-K, -(K-1), \ldots, K\}$, then we say H has spectrum K and H is universal Hamiltonian. The paper proved that by using Pauli gate multiple times, it can implement Universal Hamiltonian.

However, there is some condition.

- The spectrum K have to be large enough to approximate Fourier series.

This could be done by enlarging the size of the encoding gate

- The coefficient $\alpha_{j,k}$ have to be flexible enough.

This could be done by enlarging the universality of encoding gate and implementing any arbitrary gate. Moreover, the VQC assume that the arbitrary gate is parameterized and trained using classical optimization methods.

Universality of Quantum Kernel Method

Now, applying them into quantum kernel RKHS.

Our models with quantum kernel has the form of $\hat{f}(x) = \sum \beta_n k(x, x_n)$.

See that $k(x, x_n) = |\left\langle 0 | \phi^\dagger(x_n)\phi(x)\right|0\rangle|^2 = \left\langle 0 | \phi^\dagger(x)\phi(x_n)\right|0\rangle \left\langle 0 | \phi^\dagger(x_n)\phi(x)\right|0\rangle$.

By defining $M(x_n) = \phi(x_n)\left|0\right\rangle\left\langle 0 | \phi^\dagger(x_n)\right.$, we can see that $k(x, x_n)$ could be written as $\sum\limits_{j,k \in D} \alpha_{j,k}(x_n) e^{i(\Lambda_j - \Lambda_k)x}$.

Therefore, $\hat{f}(x) = \sum\limits_{j,k \in D} \left(\sum\limits_{n=1}^{N} \beta_n \alpha_{j,k}(x_n)\right) e^{i(\Lambda_j - \Lambda_k)x}$.

By letting $\gamma_{j,k} := \left(\sum\limits_{n=1}^{N} \beta_n \alpha_{j,k}(x_n) \right)$, we can get another Fourier series form

$$\hat{f}(x) = \sum\limits_{j,k \in D} \gamma_{j,k} e^{i(\Lambda_j - \Lambda_k)x}.$$

Therefore, the function using Quantum RKHS can make universal function approximator.

However, there is some condition.

- The spectrum K have to be high enough

This could be done by enlarging the size of the encoding gate

- The coefficient $\gamma_{j,k}$ have to be flexible enough.

This could be done by finding β_n enough. Remark that not like the VQC, $\alpha_{j,k}(x_n)$ in $\gamma_{j,k} = \left(\sum\limits_{n=1}^{N} \beta_n \alpha_{j,k}(x_n) \right)$ is fixed constant. Therefore, we need enough sample kernel $k(x, x_n)$ and find enough β_n to give the term enough expressibility.

Relation Between VQC and QKM

Commonality

- Both of them are implementing the Fourier Series directly.
- To approximate the Fourier Series with more frequency, it have to enlarge the encoding gate.
- Both of them implement the Fourier term $e^{i(\Lambda_j - \Lambda_k)x}$ with quantum gate and find Fourier coefficients $\gamma_{j,k}$ with classical methods

Difference

- Training coefficient
  - VQC
    - VQC use parameterized quantum gate and canonical optimization problem to estimate the Fourier coefficient.
    - To calculate the gradients with p-numbers of parameter with s numbers of shots, we have to run the quantum circuit $2sp$ times. To run the k number of iteration with n numbers of samples, we need to run the quantum circuit with $2nspk$ times.
  - QKM
    - QKM uses the kernel matrix and find the coefficient using classical methodology including optimization methods or eigenvalue solver.
    - Once we find the kernel matrix, we don't have to run the quantum circuit to find the coefficients. However, we have to run the quantum circuit $\frac{n(n-1)}{2}s$ times.
  - Usually, the quantity p, k is relatively smaller than n. Therefore, roughly speaking, if $2pk > n - 1$, it is more easy to run VQC vice versa. However, because the quantum circuits themselves are different and we can use more advanced methods to estimate the coefficients, we cannot precisely compare them.
  - But at least, we can see that which components make the problem harder. So according to the problem, we can chose more efficient methods.

- Evaluating new data.
  - VQC
    - To find a estimate of a data point x_{new}, VQC only need to run the circuit s times(the number of shots) and find a value $f_{\theta^*}(x_{new})$
  - QKM
    - To find a estimate of a data point x_{new}, QKM have to calculate the evaluating vector $k(x_{new}, x_1), k(x_{new}, x_2), \ldots, k(x_{new}, x_n)$. Therefore, we need to run the circuit ns times.
  - We can see that when evaluating new values, the VQC is definitely better than QKM without considering the complexity of each quantum circuit.
- Applicable.
  - VQC
    - Currently, the VQC models $\hat{f}(\theta)$ could only make the model using canonical optimization form $c(\theta) = d(f, \hat{f}(\theta))$
  - QKM
    - We can use classical methodology once we construct the Kernel Matrix. Because most of the algorithm in RKHS is built invariant of types of the kernel, we can use abundant pool of the theory. G-SIR could be example which solve the problem using Generalized Eigenvalue Problem.

Current and Potential Obstacle

1. The result of [Schuld M. (2021) The effect of data encoding on the expressive power] is little irrational. It said that a evaluation $f_\theta(x) = |\langle 0|U^\dagger(x;\theta)MU(x;\theta)|0\rangle|^2$ has a form of Fourier series $\sum \alpha_k e^{ikx}$. See that left term is real and right term is complex. There might be some situation neutralizing the imaginary term of result.
   - However, if we make the function $\langle 0|U^\dagger(x;\theta)MU(x;\theta)|0\rangle$ which could be complex, than the statement of the paper might be rational.
2. The concepts of customized kernel might not be efficient one. We'll experiment by using simulation data with various quantum kernel. The candidate of data is 1. Normal data, 2. Periodic data, 3. Deterministic Binary data, 4. Probabilistic Binary Data(Bernoulli data), 5. Data with Outlier, 6. Imbalanced data, 7. Sparse data, 8. Data with white noise, 9. Data with non-white noise. The candidate of the kernel gate is $R_y(x), R_z(x), CX, CZ$ and may be raw feature vector using $R_y(arc\cos(x))$ or other simple variation $R_u(\phi(x))$.
3. We can build theoretically well defined kernel. But it might be better to use heuristic kernel. We'll see the performance and expressibility of two kinds of kernels.

흐름

1. Introduction

2. RKHS

3. RKHS with quantum kernel

   1. positive definiteness

      2. universality

4. VQC vs QKM

5. Experiment

      1. one-line example (kernel regression)

      2. Dimension reduction with higher kernel (practical assecement)

6. Potentials

      1. customized kernel