

## 代码功能

基于昇思 MindQuantum，利用 VQA 求解一维薛定谔方程

## 方案描述

VQA(Variational quantum algorithm)是一种可在近期含噪量子设备上实现的量子、经典混合算法,在量子计算机上我们需要搭建一个浅层的含参酉电路 $U(\lambda)$ 实现相关量子态的制备,而经典计算机主要是利用量子计算机传递的关于 $C(\lambda)$ 的信息调整门序列参数 $\lambda$ ,使 $C(\lambda)$ 最小化。详细过程描述如下:

(1) 搭建浅层、含参酉门序列  $U(\lambda)$ , 输入初态  $|\psi_0\rangle$ , 经过含参酉电路后量子态为 $|\psi(\lambda)\rangle = U(\lambda)|\psi_0\rangle$ ;

(2)计算代价函数 $C(\lambda)$ ;

(3)若 $C(\lambda)$ 或迭代次数满足停止条件, 输出 $\lambda_{opt}$ , 得到 $C(\lambda_{opt})$ , 对应量子态为 $|\psi(\lambda_{opt})\rangle = U(\lambda_{opt})|\psi_0\rangle$ , 否则, 采用经典优化器(比如梯度下降)来优化更新参数 $\lambda$ , 以最小化 $C(\lambda)$ ;

(4)重复(2-3)过程, 直到 $C(\lambda)$ 或迭代次数满足停止条件;

在本项目中我们基于文献[1], 利用变分量子算法求解一维薛定谔方程, 基本方案描述如下:

$$\left[ \frac{-1}{2} \frac{d^2}{dx^2} + V(x) + g|f(x)|^2 \right] f(x) = Ef(x) \quad (1)$$

其中 $f(x)$ 在区间 $[a, b]$ 上是一个单一的实值函数, 满足归一化条件 $\int_a^b |f(x)|^2 dx = 1$ 以及周期边界条件:  $f_0 = f_N$ 。其中 $g$ 代表非线性强度,  $V(x)$ 是基本函数, 代码实现中假设 $V(x) = s_1 \sin(k_1 x) + s_2 \sin(k_2 x)$  并且  $k_2 = \frac{2k_1}{1+\sqrt{5}}$ 。

现对(1)式左侧进行积分得到(2):

$$\int_a^b f^*(x) \left[ \frac{-1}{2} \frac{d^2}{dx^2} + V(x) + g|f(x)|^2 \right] f(x) dx = K + P + I \quad (2)$$

对(1)式右侧积分得到：

$$\int_a^b f^*(x) E f(x) dx = E * \int_a^b |f(x)|^2 dx = E \quad (3)$$

根据 (1) 式可知：  $E = K + P + I$ ，其中： $K = \frac{-1}{2} \int_a^b f^*(x) \frac{d^2 f}{dx^2} dx$ ， $P = \int_a^b f^*(x) V(x) f(x) dx$ ， $I = g \int_a^b f^*(x) |f(x)|^2 f(x) dx$ 。

现将区间  $[a, b]$  离散化，划分为  $N = 2^n$  个等间距的网格点，间距  $h_N = \frac{b-a}{N}$ ， $x_k = a + k * \frac{b-a}{N}$ ，其中  $k = 0, 1 \dots N-1$ ，通过推导可知：

$$\int_a^b |f(x)|^2 dx = h_N * \sum_{k=0}^{N-1} |f_k|^2 = 1 \quad (4)$$

令  $\psi_k = \sqrt{h_N} f_k$ ，则有： $\sum_{k=0}^{N-1} |\psi_k|^2 = 1$ ，且根据  $K$ 、 $P$ 、 $I$  的表达式推导得到：

$$\llbracket K \rrbracket = -\frac{1}{2} \frac{1}{h_N^2} \sum_{k=0}^{N-1} \psi_k^* (\psi_{k+1} - 2\psi_k + \psi_{k-1}), \quad (5.1)$$

$$\llbracket P \rrbracket = \sum_{k=0}^{N-1} [\psi_k^* V(x_k) \psi_k], \quad (5.2)$$

$$\llbracket I \rrbracket = \frac{1}{2} \frac{g}{h_N} \sum_{k=0}^{N-1} |\psi_k|^4, \quad (5.3)$$

三者分别代表了动能、势能、相互作用能，三者之和等于  $E$ 。现定义代价函数如下，通过最小化代价函数  $C$  可以找到(1)的基态，即找到使得  $C$  最小的  $f(x)$ 。

$$C = \llbracket K \rrbracket_c + \llbracket P \rrbracket_c + \llbracket I \rrbracket_c \quad (6)$$

其中  $\llbracket K \rrbracket_c$ ， $\llbracket P \rrbracket_c$ ， $\llbracket I \rrbracket_c$  代表平均值。

现在问题的关键在于如何获取  $\psi_k$  的信息。通过前文内容可知  $\sum_{k=0}^{N-1} |\psi_k|^2 = 1$ ，如果将十进制下标  $k$  转化为二进制比特串  $binary(k)$ ，那么可得到  $|\psi\rangle = \sum_{k=0}^{N-1} \psi_k |binary(k)\rangle$ ，从而将  $\psi_k$  编码到量子态中。量子态  $|\psi\rangle$  可以通过含参酉序列  $U(\lambda)$  制备，记为量子态  $|\psi(\lambda)\rangle$ 。有了  $|\psi(\lambda)\rangle$  之后，可以通过构造相关量子电路实现代价函数  $C$  的计算，通过参数优化实现代价函数  $C(\lambda)$  的最小化，从而找到使得  $C(\lambda)$  最小的参数  $\lambda^*$ ，得到量子态  $|\psi(\lambda^*)\rangle = U(\lambda^*)|0\rangle$ 。根据  $\psi_k = \sqrt{h_N} f_k$  即可推出  $f(x_k)$  和  $x_k$  的对应关系，实现(1)的求解。

## 代码内函数说明

### 1. *create\_ansatz\_circuit*(*n, layer*)

参数：量子比特数*n*，要搭建的*ansatz*层数*layer*；

功能：搭建含参酉电路 $U(\lambda)$ ，实现量子态的制备 $\psi(\lambda) = U(\lambda)|0\rangle$ ；

返回：含参酉电路*ansatz\_p*；

### 2. *get\_info\_state*(*n, layer, ansatz\_data*)

参数：一维数组 *ansatz\_data*，为含参酉电路赋值；

功能：获取 $\psi(\lambda)$ 中的测量结果；

返回：字典*info*，存储十进制*k*和系数 $\psi_k$ 之间的对应关系；

### 3. *calculate\_I*(*n, layer, ansatz\_data, info*)

参数：字典*info*，存储十进制*k*和系数 $\psi_k$ 之间的对应关系

功能：计算*Interaction energies*

返回：能量值*result\_I*， *float*类型

### 4. *calculate\_K*(*n, layer, ansatz\_data, info*)

参数：意义同上，略

功能：计算*Kinetic energies*

返回：能量值*result\_K*， *float*类型

### 5. *calculate\_P*(*n, layer, ansatz\_data, info*)

参数：意义同上，略

功能：计算*Potential energies*

返回：能量值*result\_P*， *float*类型

### 6. *cost\_function*(*n, layer, ansatz\_data*)

参数：意义同上，略

功能：计算代价函数值，上述三个能量值之和

返回:  $result\_K, result\_P, result\_I$  以及  $function\_value$ ,  $float$  类型

7. *calculate\_gradient(n, layer, ansatz\_data)*

参数:  $layer$  是当前  $ansatz$  层数

功能: 利用有限差分法计算梯度, 存储在列表  $gradient$  内

返回: 列表  $gradient$  以及在  $ansatz\_data$  下的函数值  $function\_value$

8. *global\_training(n, p)*

参数: 量子比特数  $n$  以及  $ansatz$  总层数  $p$

功能: 一次性搭建  $p$  层  $ansatz$  电路, 并且实现参数随机初始化、调用  $scipy$  内的  $minimize$  函数进行参数优化,

返回: 迭代停止时的参数  $ansatz\_data$

9. *layerwise\_training(n, p)*

参数: 意义同上, 略

功能: 逐层搭建  $ansatz$  电路, 直至电路层数为  $p$ , 同时实现参数优化

返回: 电路为  $p$  层时, 代价函数收敛时对应的参数  $ansatz\_data$  以及  $function\_value$

10. *train\_parameters(layer, counts, ansatz\_data)*

参数:  $ansatz\_data$  为电路层数为  $layer - 1$  层时, 代价函数收敛时对应的参数, 其中  $1 \leq layer \leq p$ ;

功能: 实现前  $layer - 1$  层参数的赋值以及第  $layer$  层参数的随机初始化, 利用  $scipy$  内  $minimize$  函数实现全体参数的优化

返回: 电路为  $layer$  层, 代价函数收敛时对应的参数  $ansatz\_data$  以及  $function\_value$

**其余需要说明的问题:**

利用有限差分法获取梯度信息的时候, 代码中的  $delta$  虽然是写死的, 但是这是

一个超参数，需要调节的，否则可能会提示“Desired error not necessarily achieved due to precision loss”，目前暂未想到获取精确梯度信息的好办法，只能通过有限差分获取近似梯度，大家如果有其余获取梯度信息的好办法，也欢迎及时评论，将及时进行代码的相关完善。

补充：本问题内哈密顿形式无法写成 $pauli$ 积张量的形式，无法利用 mindquantum 下的`get_expectation_grad()`获取梯度信息。

### ***Fixing strategy***思想介绍

*Fixing strategy*策略编码在`train_parameters(layer, counts, ansatz_data)`以及`layerwise_training(n, p)`中，下面为*Fixing strategy*的基本思想：

假设搭建 $p$ 层 $ansatz$ 电路， $layer$ 代表当前  $ansatz$  层数，每层  $ansatz$  电路有 $m$ 个参数

(1)  $layer = 1$ ,  $\vec{\lambda} = (\vec{\lambda}_1) = (\lambda_1, \lambda_2 \dots \lambda_m)$ , 随机初始化 $\vec{\lambda}$ ,  $\lambda_{1\dots m} \in [0, 2\pi]$ , 优化 $\vec{\lambda}$ , 得到优化结果 $loss$ ;

(2) 重复(1) $counts$ 次，得到 $counts$ 个 $loss$ ，将 $\min(loss)$ 对应的优化之后的参数记为 $\vec{\lambda}_1^*$ ;

(3)  $layer = 2$ ,  $\vec{\lambda} = (\vec{\lambda}_1^*, \vec{\lambda}_2)$ , 其中 $\vec{\lambda}_2$ 随机初始化，优化 $\vec{\lambda}$ ，得到优化结果 $loss$ ;

(4) 重复(3) $counts$ 次，得到 $counts$ 个 $loss$ ，将 $\min(loss)$ 对应的优化之后的参数记为 $(\vec{\lambda}_1^{**}, \vec{\lambda}_2^*)$ ;

(5)  $layer = 3$ ,  $(\vec{\gamma}, \vec{\beta}) = (\vec{\lambda}_1^{**}, \vec{\lambda}_2^*, \vec{\lambda}_3)$ , 其中 $\vec{\lambda}_3$ 随机初始化，优化 $\vec{\lambda}$ ，得到优化结果 $loss$ ;

(6) 重复(5) $counts$ 次，得到 $counts$ 个 $loss$ ，将 $\min(loss)$ 对应的优化之后的参数记为 $(\vec{\lambda}_1^{***}, \vec{\lambda}_2^{**}, \vec{\lambda}_3^*)$ ;

(7)重复上述过程, 直到 $layer = p \dots$

## 参考文献

- [1] Lubasch, M., Joo, J., Moinier, P., Kiffner, M., & Jaksch, D. (2020). Variational quantum algorithms for nonlinear problems. *Physical Review A*, 101(1).
- [2] Lee, X, et al. "Parameters Fixing Strategy for Quantum Approximate Optimization Algorithm." *IEEE International Conference on QCE*(2021).