From QCNN to QAMCNN

```
QCNN is not the CNN in classical sense, where QAMCNN tries to be.
```

Q0: 我们想用纯量子神经网络方法挑战诸如 cifar10 等图像分类任务

基本框架:

```
                    [B, D]      [B, NC]
|0>--|         |--|        |--|         |
...--| encoder |--| ansatz |--|  pauli  |
|0>--|         |--|   (θ)  |--| measure |--CELoss(softmax)--
```

编码器 encoder: 由于 cifar10 图像较大，我们不打算下采样后再 AngleEncode，直接考虑 AmplitudeEncode
线路拟设 ansatz: Hardware-Efficient Ansatz 效果应该是不好的，我们直接考虑**看似最合理**的 QCNN

接下来就是故事、问题、分析 ↓↓↓

Q1: 为什么对于 QCNN 而言，AmpEnc 可能不是合理的输入数据结构

以 `4x4` 标准顺序平铺的图像像素矩阵 M_{ij} 为例:

```
i\j    0      1      2      3
 0 | 0000 | 0001 | 0010 | 0011 |
 1 | 0100 | 0101 | 0110 | 0111 |
 2 | 1000 | 1001 | 1010 | 1011 |
 3 | 1100 | 1101 | 1110 | 1111 |
```

将数据编码到 4-qubit 量子态振幅上，一般而言即:

$$|\psi\rangle = \sum_{k=0}^{15} a_k * |k\rangle$$

其中 a_k 为矩阵 M_{ij} 中各项系数，使用**行优先**存储，k 为下标索引的二进制串形式

而 QCNN 的结构一般如下 (小端序，上方低位下方高位):

```
          (conv1) (pool1) (conv2) (pool2)
|0>---------|---|--|---|
            | U |  | V |
|0>--|---|--|---|--|---|--|---|--|---|
     | U |                |   |  |   |
|0>--|---|--|---|--|---|  | U |  | V |
            | U |  | V |  |   |  |   |
|0>---------|---|--|---|--|---|--|---|--M
```

现简化考虑 QCNN 中所用的 U 门结构 (或即任何 SU(4) 族酉阵的成员) 为一个万能 TwoLocal 门，即作用在相邻比特上的通用两比特门 U，我们先看一个 U 门的作用：

$$
\begin{aligned}
|\phi\rangle &= (I^{\otimes 2} \otimes U)\,|\psi\rangle \\
&= (I^{\otimes 2} \otimes U)\,|abcd\rangle \\
&= |ab\rangle \otimes U\,|cd\rangle \\
&= |ab\rangle \otimes (\alpha_0\,|00\rangle + \alpha_1\,|01\rangle + \alpha_2\,|10\rangle + \alpha_3\,|11\rangle) \\
&= \alpha_0\,|ab00\rangle + \alpha_1\,|ab01\rangle + \alpha_2\,|ab10\rangle + \alpha_3\,|ab11\rangle
\end{aligned}
$$

可见此操作：

- 在比特意义上: 局部比特相互作用，打散或整合，不影响其他比特
- 在振幅意义上: 区分出 4 组，振幅重分配
- **在图像意义上: 按列加权 (或筛选)，整体地 增加/衰减 某一列的值**

那么两个并排的 U 门呢？

$$
\begin{aligned}
|\phi\rangle &= (U \otimes U)\,|abcd\rangle \\
&= U\,|ab\rangle \otimes U\,|cd\rangle \\
&= (\alpha_0\,|00\rangle + \alpha_1\,|01\rangle + \alpha_2\,|10\rangle + \alpha_3\,|11\rangle) \otimes (\beta_0\,|00\rangle + \beta_1\,|01\rangle + \beta_2\,|10\rangle + \beta_3\,|11\rangle) \\
&= (\alpha_0 * Row_0 + \alpha_1 * Row_1 + \alpha_2 * Row_2 + \alpha_3 * Row_3) \otimes (\beta_0 * Col_0 + \beta_1 * Col_1 + \beta_2 * Col_2 + \beta_3 * Col_3)
\end{aligned}
$$

也就是行-列同时各自成组地重分配了振幅 =_=||

也就是说前半段比特负责行、后半段负责列，那么上述 QCNN 图示中的第一个 U 门显然就可以同时操作来自行-列的信息了：

$$
\begin{aligned}
|\phi\rangle &= (I \otimes U \otimes I)\,|abcd\rangle \\
&= |a\rangle \otimes U\,|bc\rangle \otimes |d\rangle \\
&= |a\rangle \otimes (\alpha_0\,|00\rangle + \alpha_1\,|01\rangle + \alpha_2\,|10\rangle + \alpha_3\,|11\rangle) \otimes |d\rangle \\
&= \alpha_0\,|a00d\rangle + \alpha_1\,|a01d\rangle + \alpha_2\,|a10d\rangle + \alpha_3\,|a11d\rangle
\end{aligned}
$$

注意到这四项分别对应 M_{ij} 中的四个 `stride=2` 的子矩阵，也就是说这个操作接近于 PixelShuffle/PixelUnshuffle 或 JPEG 原理，而非经典 CNN 局部滤波模型

```
      |a00d>                   |a01d>
 | * |   | * |   |          |   | * |   | * |
 |   |   |   |   |          |   |   |   |   |         ...
 | * |   | * |   |          |   | * |   | * |
 |   |   |   |   |          |   |   |   |   |
```

因此从 AmpEnc 出发，QCNN 的 conv+pool 组合实际在对每个 **稀疏子图** 和 **行-列** 进行振幅重分配，随后丢弃偶数行-列，不断重复此过程直到剩余子图足够小，然后测量...

- 一定意义上这种计算结构可以工作，如果它能通过振幅重分配
  - 将重要的判别性信息转移到奇数行列（因此偶数行列可以安全舍弃）
  - 调整这些行-列的整体权重，亦即相对重要性 (从而映射到判别结果/score/logits)
- 但这种计算结构的建模意义可能比较奇怪

- 提取的图像"特征"总是某行或某列这个整体，无法抽取图像二维意义上的局部信息
- 须注意: AmpEnc+QCNN 和经典 CNN 局部滤波模型在数学上完全不等价！！
  - 破坏了图像二维的邻域性
  - 也不能保证平移不变性

Q2: 那么 AmpEnc 更合理的用法是什么呢

我们认为毫无疑问是 PREPARE-SELECT 结构!

引入一个辅助寄存器 `|anc>`，我们可以通过多比特控制门，把振幅编码寄存器 `|enc>` 里的东西读出来，而**不会就地破坏原始数据**:

```
|enc>  PREP           SEL         PREP†
 |0>--|     |--o--●------...--●--|     |
 |0>--| amp |--o--o------...--●--| amp |
 |0>--| enc |--o--o------...--●--| enc |
 |0>--|     |--o--o------...--●--|  †  |
               |  |           |
|anc>          |  |           |
 |0>-----------RY-|-----------|----------
 |0>--------------RY----------|----------
 ...                          |
 |0>-------------------------RY---------
```

不难验证，当 `|enc>` 寄存器中**所有**比特都受控时，对应的 `RY` 门恰好把振幅编码中对应下标索引的单个分量读出到 `|anc>` 寄存器中，以控制位 `0010`、转角 $RY(\pi)$ 为例，即:

$$
\begin{aligned}
|\phi\rangle &= \text{mctrl}(RY, 0010)\,|anc\rangle\,|enc\rangle \\
&= \text{mctrl}(RY, 0010)\,|0...00\rangle \otimes \sum_{k=0}^{15} a_k * |k\rangle \\
&= |0000000000000\rangle\,(RY(\pi)\,|0\rangle)\,|00\rangle \otimes a_2\,|0010\rangle + \text{others} \\
&= |0000000000000100\rangle \otimes a_2\,|0010\rangle + \text{others} \\
&= a_2\,|0000000000000100, 0010\rangle + \sqrt{1-a_2}\,|0...0\rangle \otimes \sum |abcd\rangle
\end{aligned}
$$

这就 (逻辑意义上) 完美地把第2个像素的值 (从0起计数)，即矩阵系数 a_2 从 `|enc>` 寄存器转移到了 `|anc>` 寄存器，此时仅通过访问 `|anc>` 寄存器就可以取到这个本在 `|enc>` 中的数据了:
单看 `|anc>` 中的2号比特，其振幅为 $|anc_2\rangle = a_2\,|1\rangle + \sqrt{1-a_2}\,|0\rangle$

ℹ 注意到这个转换是一种 AmplitudeEncode → AngleEncode
⚠ 在 NISQ 时代，`|anc>` 寄存器可能不够大，以至于无法完全解压出 `|enc>` 中的信息!

那么扩展一下，如果 `|enc>` 寄存器中只有**部分**比特受控呢:

```
|enc>  PREP     SEL     PREP†
 |0>--|     |-----..--|     |
 |0>--| amp |--○--..--| amp |
 |0>--| enc |--●--..--| enc |
 |0>--|     |--|--..--|  †  |
               |
|anc>          |
 |0>-----------RY------------
```

显然它选定的是比特串模式 `|x10x>` ，其中 `x` 表示 0/1 均可，也即有四项振幅会影响 `|anc>` 中的辅助比特旋转情况，反过来说即，通过该辅助比特可以取到某四项振幅的线性组合；该例模式对应图像上的像素位置为：

```
|   |   |   |   |
| x | x |   |   |
|   |   |   |   |
| x | x |   |   |
```

注意到它有点像，但并不是 QCNN 里的 **稀疏子图** 选择器，并且看起来只需要一点点额外的设计，它就能实现和经典 CNN 一样的 **局部块选择**了！！

Q3: 如何让 AmpEnc 上的 PREP-SEL 能像 CNN 一样选择到局部块呢

答案是 QAM 星座图；我们只需要改变从二维图像平坦转为一维向量时**像素读取的先后顺序**即可！

考虑如下 QAM16 阵列，借助其格雷码的性质，任意一个 `2x2` 的局部子块都有一个简短的索引：

```
i\j    0      1      2      3
 0 | 1011 | 1001 | 0001 | 0011 |
 1 | 1010 | 1000 | 0000 | 0010 |
 2 | 1110 | 1100 | 0100 | 0110 |
 3 | 1111 | 1101 | 0101 | 0111 |
```

简短索引即单个比特模式串/单个多比特控制门控制模式序列，此例中四个角上的矩阵分别是：`|10xx>` ，`|00xx>` ，`|11xx>` ，`|01xx>` ，中央矩阵是 `|xx00>` ，还有各边居中的四个分别为 `|1xx0>` ，`|x10x>` ，`|0xx0>` ，`|x00x>`

当扩展到更大的 QAM 矩阵时，所以**并非**所有边长为 `2^k` 的矩阵都有单一简短索引，但大部分经典 CNN 中所关心的块位置是可以取到简短索引到的，这暗示我们也许**有希望在量子神经网络模型里复刻经典 CNN 的数学结构**！

Q4: pseudo-QAMCNN: QAM顺序真的比朴素的行扫描好吗

标准 QCNN 搭配 QAM 平坦化策略，在数学上会有什么区别呢？

还是先来看一个 U 门做了什么 (注意我们先把 U 放在高位比特上)：

$$
\begin{aligned}
|\phi\rangle &= (U \otimes I^{\otimes 2})\,|\psi\rangle \\
&= (U \otimes I^{\otimes 2})\,|abcd\rangle \\
&= U\,|ab\rangle \otimes |cd\rangle \\
&= (\alpha_0\,|00\rangle + \alpha_1\,|01\rangle + \alpha_2\,|10\rangle + \alpha_3\,|11\rangle) \otimes |cd\rangle \\
&= \alpha_0\,|00cd\rangle + \alpha_1\,|01cd\rangle + \alpha_2\,|10cd\rangle + \alpha_3\,|11cd\rangle \\
&= \alpha_0\,|\nearrow\rangle + \alpha_1\,|\searrow\rangle + \alpha_2\,|\nwarrow\rangle + \alpha_3\,|\swarrow\rangle
\end{aligned}
$$

可见此操作在图像意义上：

- 选出了右上、左下、左上、右下的四个 `2x2` 小块
- 重分配了这四块的振幅，某种意义上是一个 spatial attn!!

那么两个并排的 U 门呢？

- 分析可知低比特上的 U 门同样选出了四小块，但是不具备图像像素连续性，没用、甚至是坏的
- 标准 QCNN 线路设计可能**不直接匹配**我们的新数据结构

总地来说线路的工作方式变化 (稀疏子块/行/列 → 局部块)

- conv: 选择一些局部块，重分配振幅
- pool: 丢弃掉一半的局部块

但我们做了消融实验，qam 比 std 稳定地好一点，也确实只能好一丁点 (1~2个百分点的量级)

flat	enc	clf	train acc	test acc	comment
std	AmpEnc	QCNN			基线方法
qam	AmpEnc	QCNN			过渡性的优化解决方案
qam	AmpEnc	QAMCNN			这才是我们最终的目标！！

ℹ 我们认为这主要是 QCNN 的锅，而不是 QAM，应该着手设计匹配的线路 ((逃

Q5: QAMCNN: QAM 平坦策略加上基于 PREP-SEL 结构的线路，能做出来数学上等价的 CNN 吗

我们相信在**理论**上应该是可以逼近的！
纲领：我们的计算一定要以振幅为中心，信息编码到振幅，计算也是幅值上的算数运算，最终结果也出现在某些项的振幅之中

目前限制我们继续推进设计-实现的主要因素是经典计算机无法模拟更大的 `|anc>` ，存不下太大的 feature map，也就无从谈论在**实践**上对标经典 CNN——毕竟我们的经典内存已经太便宜了，空间复杂度很少被考虑...

by Armit
2024年11月12日