A QCNN for Quantum State Preparation

Carnegie Vacation Scholarship

David Amorim

Weeks 7-8
(12/08/2024 - 23/08/2024)

Aims for the Week

The following aims were set at the last meeting (14/08/2024):

New Phase Encoding Approach

Investigate a new approach to phase encoding using linear piecewise phase functions without explicit function evaluation.

Handover

Hand over the slides, documentation, code and the poster for the Carnegie Trust.

Table of Contents

Preliminaries

- Consider an n-qubit register with computational basis states $|j\rangle = |j_0 j_1 ... j_{n-1}\rangle$ representing n-bit strings
- Let p of the register qubits be precision qubits so that

$$j = \sum_{k=0}^{n-1} j_k 2^{k-p} \tag{1}$$

- Consider a phase function Ψ over the domain $\Omega = \{j\}$ and construct an M-fold partition ($M = 2^g$, $g \leq n \in \mathbb{N}$) into equal sub-domains Ω_m:

$$\Omega = \bigcup_{m=1}^{M} \Omega_m, \quad \Omega_m \cap \Omega_l = \emptyset, \quad |\Omega_m| = |\Omega_l| \tag{2}$$

- On each sub-domain, approximate Ψ using a linear function:

$$\Psi(j) = \alpha_m j + \beta_m, \quad j \in \Omega_m \tag{3}$$

Phase Encoding within a Sub-domain

Aim 1

For $j \in \Omega_m$ construct an operator $\hat{O}_m$ such that $|j\rangle \mapsto e^{i(\alpha_m j + \beta_m)} |j\rangle$.

- Consider the single-qubit operators

$$\hat{P}^{(k)}(\varphi) = \begin{pmatrix} e^{i\varphi} & 0 \\ 0 & e^{i\varphi} \end{pmatrix}, \quad \hat{R}^{(k)}(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix} \tag{4}$$

  each acting on the kth qubit

- Then

$$\hat{O}_m \equiv \bigotimes_{k=0}^{n-1} \hat{P}^{(k)}(\beta_m/n) \hat{R}^{(k)} \left(\alpha_m 2^{k-p} \right) \tag{5}$$

  transforms

$$|j\rangle \mapsto \exp\left[i \left(\sum_{k=0}^{n-1} \alpha_m j_k 2^{k-p} + \beta_m \right) \right] |j\rangle = e^{i(\alpha_m j + \beta_m)} |j\rangle \tag{6}$$

Selecting the Subdomain

- It is straight-forward to construct $\hat{O}_m$ for each of the sub-domains Ω_m
- More challenging is applying the correct $\hat{O}_m$ based on the sub-domain corresponding to each $|j\rangle$

Aim 2

Construct a system of controls such that $\hat{O}_m$ is applied to $|j\rangle$ if and only if $j \in \Omega_m$

Sample Case: $M = 2$

- Start with the simplest possible case, a 2-fold partition ($M = 2$):

$$j \in \begin{cases} \Omega_1 & j_0 = 0 \\ \Omega_2 & j_0 = 1 \end{cases} \quad (7)$$

- Using an ancilla qubit, $\hat{O}_1$ is applied for $j \in \Omega_1$ and $\hat{O}_2$ for $j \in \Omega_2$
- The ancilla is required since the operation applied to $|j_0\rangle$ is conditional on $|j_0\rangle$ itself

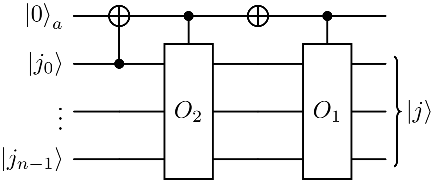

Figure 1: Circuit diagram for $M = 2$

Generalising the Approach

- The approach shown on the previous slide requires $1 \leq \log_2 M \leq n$ ancilla qubits
- The number of controls required is $\sim \mathcal{O}(M \log M)$ as there are M operators, each controlled by all ancillas
- A 'pyramid' network of X gates is applied to the ancillas for case distinction
- For $M \sim 2^n$ the gate cost is exponential!

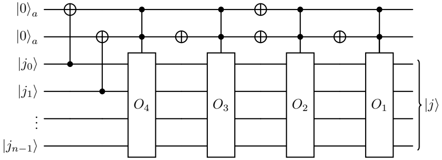

Figure 2: Circuit diagram for $M = 4$. Note that $j \in \Omega_m$ if $j_0 j_1 = m - 1$ (e.g. $j \in \Omega_3$ if $j_0 j_1 = 10$).

A Recursive Approach

- Since $M = 2^g$ for some $g \leq n \in \mathbb{N}$ we can view the partition of Ω as a recursive process, splitting the domain into halves g times
- ....

TRY TO GET RID OF ANCILLAS!! THIS IS ESSENTIALLY A CLASSICAL ALGORITHM! MUST BE POSSIBLE TO DO IT BETTER!!

Qiskit Implementation

Table of Contents

Handover

The code, documentation, slides, and poster are all available on GitHub:

`https://github.com/david-f-amorim/PQC_function_evaluation`

- The source code is found in the directory `pqcprep`
- The slides and poster are found in the directory `slides`
- The documentation is hosted externally here, which is linked on GitHub