

# Intorduction to Quantum Computing: Homework #1

Due on Star Wars Day 2020

Jakub Filipek

## Problem 1

### Part (a)

There are exactly  $n$  Hadamard gates and  $n^2 R_Z$  gates in a QFT circuit.

I will assume that Hadamards are implemented in an errorless fashion (this is unrealistic, but the problem, does not specify bounds on Hadamard gate error).

Lastly, let's note that when gates are applied the error scales linearly with number of gates applied.

$$\begin{aligned}\|F_{2^n} - \tilde{F}_{2^n}(\theta)\| &\leq \|n^2 R_Z(\theta) - n^2 \tilde{R}_Z(\theta)\| \\ &= n^2 \|R_Z(\theta) - \tilde{R}_Z(\theta)\| \\ &\leq n^2 \Delta\end{aligned}$$

Since  $\Delta \in O(\frac{\epsilon}{n^2})$ , we can choose set  $\epsilon = \frac{1}{k} n^2 \Delta$ , where  $k > 1$  (this is stronger than standard big- $O$  requirement).

This means that we can choose  $\epsilon$  to be tuned (by choosing  $k$ ) to be exactly:

$$\begin{aligned}\|F_{2^n} - \tilde{F}_{2^n}(\theta)\| &= \epsilon \\ &= \frac{1}{k} n^2 \Delta \\ &\leq n^2 \Delta\end{aligned}$$

Hence such implementation is possible.

### Part (b)

Firstly let us note that an Adder contains 2 QFT circuits, and an additional set of  $n^2 R_Z$  gates. Hence, there is total of  $3n^2 R_Z$  gates.

$$\begin{aligned}\|ADD - \widetilde{ADD}\| &= \|3n^2 R_Z - 3n^2 \tilde{R}_Z\| \\ &= 3n^2 \|R_Z - \tilde{R}_Z\| \\ &\leq 3n^2 \Delta \\ &\leq 3n^2 \frac{\epsilon}{3n^2} \\ &= \epsilon\end{aligned}$$

Using that  $\Delta \in O(\frac{\epsilon}{n^2})$ .

Then:

$$\|ADD - \widetilde{ADD}\| \leq \epsilon$$

For the second part  $\Delta \in O(\frac{\epsilon}{n^2})$ , implies that  $\epsilon \geq \frac{1}{k}\Delta n^2$ , and since  $n$  can be arbitrarily big, while  $0 < \Delta \leq 2$ , the  $k$  would have to be  $\propto \frac{1}{n^2}$  for  $\epsilon < 1$ , which I do not know how to prove in such case.

### Part (c)

Firstly let us note that worst case implementation of  $\widetilde{ADD}$  will have error of  $\epsilon$  in the state space.

Now let us consider, what would be the worst case impact of error  $\epsilon$  in the state space on the measured number. Since, both  $|x\rangle$  and  $|y\rangle$  have  $n$  bits, their most significant bit represents power of  $2^{n-1}$ .

This also represents the highest error in the number space, since if we make a mistake we would be  $2^{n-1}$  away from true answer. Hence, let:

$$ADD|x\rangle|y\rangle \rightarrow |x\rangle|a\rangle$$

$$\widetilde{ADD}|x\rangle|y\rangle \rightarrow |x\rangle|\tilde{a}\rangle$$

For discrete gradients the variance within a range can be maximized if there are only 2 observations possible, but they are maximally apart. Hence let:

$$|a\rangle = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad |\tilde{a}\rangle = \begin{pmatrix} 0 \\ \vdots \\ 1 - \frac{\epsilon}{2} \\ \vdots \\ \frac{\epsilon}{2} \\ \vdots \\ 0 \end{pmatrix}$$

Where  $1 - \frac{\epsilon}{2}$  in  $|\tilde{a}\rangle$  is at the same index as the 1 in  $|a\rangle$ , and  $\frac{\epsilon}{2}$  is  $2^{n-1}$  indexes away from the previously mentioned elements (this can be either above or below them).

Without loss of generality let us assume that  $z$  is the index of  $1 - \frac{\epsilon}{2}$ , and that it's  $\leq 2^{n-1} - 1$ . This way we can assume that index of  $\frac{\epsilon}{2}$  is  $z + 2^{n-1}$ . In general, if  $z$  does not satisfy this bound we can just switch the computation below with minus sign, which doesn't change the overall bound.

Hence, the overall variance will be:

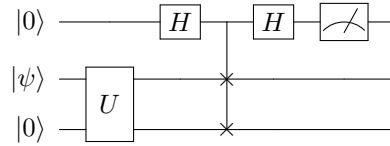
$$\begin{aligned} Var(\widetilde{ADD}) &= \sum_x P(x)f(x)^2 - (P(x)f(x))^2 \\ &= ((1 - \frac{\epsilon}{2})z^2 - ((1 - \frac{\epsilon}{2})z)^2) + ((\frac{\epsilon}{2})(z + 2^{n-1})^2 - (\frac{\epsilon}{2}(z + 2^{n-1}))^2) \\ &= z^2((1 - \frac{\epsilon}{2}) - (1 - \epsilon + \frac{\epsilon^2}{4})) + (z + 2^{n-1})^2(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}) \\ &= z^2(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}) + (z + 2^{n-1})^2(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}) \\ &= (z^2 + (z + 2^{n-1})^2)(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}) \\ &= (2z^2 + z2^n + 2^{2n-2})(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}) \end{aligned}$$

Large values of  $z$  maximize the above equation. However, since it is bounded by  $2^{n-1} - 1$ :

$$\begin{aligned}
\text{Var}(\widetilde{ADD}) &= (2z^2 + z2^n + 2^{2n-2})\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&\leq (2(2^{n-1} - 1)^2 + (2^{n-1} - 1)2^n + 2^{2n-2})\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&= (2(2^{2n-2} - 2^n + 1) + 2^{2n-1} - 2^n + 2^{2n-2})\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&= (2^{2n-1} - 2^{n+1} + 2 + 2^{2n-1} - 2^n + 2^{2n-2})\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&= (2^{2n} + 2^{2n-2} - 2^n - 2^{n+1} + 2)\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&\leq (2^{2n} + 2^{2n-2})\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&\leq (2 \cdot 2^{2n})\left(\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right) \\
&= 2^{2n}\left(\epsilon - \frac{\epsilon^2}{2}\right) \\
&\leq 2^{2n}\left(\epsilon + \frac{\epsilon^2}{4}\right) \\
&\leq 2^{2n}4\epsilon\left(1 + \frac{\epsilon}{4}\right)
\end{aligned}$$

### Part (d)

Let  $U \in \{ADD, \widetilde{ADD}\}$ . Then for an arbitrary state  $|\psi\rangle$  we can perform a modified swap test:



Note that the part after  $U$  gate is just a SWAP test. There are two cases:

- If  $U = ADD$  then, after  $U$  we will have state  $|0\rangle|\psi\rangle|\psi\rangle$ , which, after swap test will result in  $P(0) = 1$
- If  $U = \widetilde{ADD}$  then, after  $U$  we will have state  $a|0\rangle|\psi\rangle|\psi\rangle + b|0\rangle|\psi\rangle|\epsilon'\rangle$ , where length of  $|\epsilon'\rangle$  is at most  $\epsilon$ . Note that  $|\epsilon'\rangle$  is orthogonal to  $|\psi\rangle$ , since if there exists a component of it which is proportional to  $|\psi\rangle$  we can just include it in the  $|\psi\rangle$  hence increase ratio of  $a$  to  $b$  (described in next sentences). Hence  $a = \frac{||\psi\rangle|}{\sqrt{||\psi\rangle|^2 + ||\epsilon'\rangle|^2}} \geq \frac{1}{\sqrt{1+\epsilon^2}}$ . Similarly  $b \leq \frac{\epsilon}{\sqrt{1+\epsilon^2}}$ . Hence in this case  $P(0) \geq \frac{1}{2} + \frac{1+\epsilon\langle\epsilon'|\psi\rangle}{2\sqrt{1+\epsilon^2}} = \frac{1}{2} + \frac{1}{2\sqrt{1+\epsilon^2}}$ .

Note that, unless  $\epsilon = 0$  (i.e. there is no error), then  $U = ADD$  is more probable, since  $\frac{1}{2} > \frac{1}{2\sqrt{1+\epsilon^2}}$ . This can be directly seen from Bayes Theorem:

$$\begin{aligned}
P(ADD|0) &= \frac{P(0|ADD)P(ADD)}{P(0|ADD)P(ADD) + P(0|\widetilde{ADD})P(\widetilde{ADD})} \\
&= \frac{P(ADD)}{P(ADD) + \left(\frac{1}{2} + \frac{1}{2\sqrt{1+\epsilon^2}}\right)P(\widetilde{ADD})} \\
&= \frac{2}{3 + \frac{1}{\sqrt{1+\epsilon^2}}} \quad \text{Assuming: } P(ADD) = P(\widetilde{ADD}) \\
&\in \left[\frac{1}{2}, \frac{2}{3}\right] \quad \text{For: } 0 \leq \epsilon \leq 1
\end{aligned}$$

In the worst case scenario (for distinguishing between the two operations), there is no error (which means the implementation is great). Hence, the  $P(0|ADD) = P(0|\widetilde{ADD})$ , and  $P(ADD|0) = \frac{1}{2}$  (under above assumptions). Then our best option is a random guess and our success rate is 50%.

In the best case, and worst implementation:  $P(ADD|0) = \frac{2}{3}$ . This means that we should guess ADD if we see 0, and  $\widetilde{ADD}$  if we see 1. In that case our success rate is 66%.

## Problem 2

For the project I thought of implementing a Quantum Reinforcement Learning or Quantum Gradient Descent.

- The idea for Reinforcement Learning was mentioned in a talk at [Quantum Machine Learning Workshop in 2018](#) called "A Route towards Quantum-Enhanced Artificial Intelligence" by Vedran Dunjko.

I am planning on reproducing the experiment in a relatively simple classical environment with a quantum agent. This way I would be able to learn more in depth about Qubitization and the Grover Search, as well as present an interesting application of both of these techniques in machine learning.

Hence the problem would be to teach an agent such that it can efficiently solve a set of steps Maze as described in [the talk at around 26 mins](#). This should be a system small enough to efficiently simulate, while showing a lot of edge cases.

- The Gradient Descent is something I believe would be extremely interesting to do in conjunction to the *qml* project, since it would allow for an additional place to have quantum computing accelerate learning.

The idea is basically outlined in [this talk](#) starting at about 23rd minute. I would attempt to implement [Jordan's algorithm](#) and compare it to a vanilla classical SGD. The goal would be to try to learn a simple convex function (maybe with a local minimum), and then try to introduce some noise to the function itself, maybe a non-ideal black-box query.

If time allows I would try to improve it by first including improvements to the gradient descent by using Grover Search.

I mostly choose this problem, because quantum gradient descent is something that I wanted to learn and implement from start of the research project, and it seems that the final project for this class would be an ideal moment to do it. That being said I might be significantly underestimating amount of time that goes into either of these projects.

Personally, I would prefer to try doing Gradient Descent, mostly because I can ask you directly for help (since it was your talk). However, I would like to hear your feedback on whether it is a feasible final project.

As of the code, I am planning on writing it in either numpy (using linear algebra), qiskit or Q#. Numpy would definitely be the simplest to modify and customize, while other two would provide a more-realistic quantum interface, with qiskit possibly allow for actual tests on free tier of IBM Q.