Quantum Machine Learning: A Case Study of Grover's Algorithm

Bikram Khanal* , Pablo Rivas† , *Senior, IEEE*, Javier Orduz ‡ Alibek Zhakubayev §
School of Engineering and Computer Science
Department of Computer Science, Baylor University
Email: {*Bikram_Khanal1, †Pablo_Rivas, ‡Javier_OrduzDucuara, §Alibek_Zhakubayev1 }@Baylor.edu

***Abstract*—The complexity of searching algorithms in classical computing is a perpetual researched field. Quantum computers and quantum algorithms can compute these problems faster, and, in addition, machine learning implementation could provide a prominent way to boost quantum technology. We call quantum machine learning to this novel set of tools coming from artificial intelligence and quantum mechanics. To achieve our purpose, we focus on applications on quantum machine learning; in particular, we propose a review and exploration of topics such as variational quantum algorithms, kernel methods, and a review of Grover's algorithm (GA) as a quantum classifier. We start with the GA exploration to achieve this goal, which is a quantum search algorithm that achieves quadratic speedup over optimal classical search implementation. This paper implements a GA exploration that includes a concept summary and implementation, considering only AND, XOR, and OR gates. We also discuss potential in quantum machine learning.**

***Index Terms*—quantum machine learning, Grover's algorithm, classification**

I. Introduction

The searching problem to locate the distinct record from a sizeable unstructured dataset with $N = 2^n$ data points is often phrased intricacy [1]. For example, if we want to find the shortest route in a map such that the route passes through all the cities, we can search for all the possible routes and only keep those routes with the shortest path. Classically, we can search for a record x from a database in $N/2$ tries and N tries in the worst scenario. The probability theory claims that given x records, we can obtain the particular record a in $O(N)$ complexity. We will discuss in detail that this problem can be solved in $O(\sqrt{N})$ using quantum algorithms [2], [3].

In recent years, quantum algorithms are proven to speedup the complexity to its classical counterparts. Deutsch's Algorithm was the the first quantum algorithm that performed faster than its classical one [4]. In 1994 Shor's algorithm achieved exponential speedup than any classical algorithms for factorization and finding discrete logarithms. Grover proposed a searching quantum algorithm that is polynomially faster than optimal classical searching algorithms [5]. GA is a search algorithm that has applications on physics (collision problems), computational complexity (NP-complete problems) and computing (searching unsorted databases). We can use Grover's algorithm in above mentioned finding shortest route problem and any unsorted database searching problem and achieve the polynomial speedup in computation [5]. In this paper, we aim to provide mathematical Grover's Algorithm analysis, and implementation, of a particular version of its classical version.

Quantum algorithms, as those mentioned previously, implement transformations which are matrices, in terms of Linear Algebra, and those have their particular (Dirac) representation [6]. Each transformation requires an operator to create superposition, rotation (on system or on state), or another change on the system. Operator act on states. Each of those states has the form $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, which is the general state in the $\{|0\rangle, |1\rangle\}$ basis, where $\{\alpha, \beta\}$ are complex numbers, a.k.a. amplitudes [7]. In quantum circuit model, one operator, namely it acting on states, has the form:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{1}$$

$$H = \frac{1}{\sqrt{2}}\Big(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1| \Big). \tag{2}$$

Equation (1) is the matrix representation; and eq. (2) is the Dirac (*bracket*) representation. H transforms single-qbit as $|\psi\rangle$. Nonetheless, we can propose a generalization, it means if we apply H on each qbit in a system with N qbits, the system is now on superposition. Grover use this property along with oracle(a black box) and diffuser to search a unsorted database in $O(\sqrt{N})$ complexity. We can implement this circuit to reduce the runtime of any database search by running it on a quantum device.

With the algorithm shown in Figure 1, the goal is to find w, given an oracle U_f with

$$f : \{0,1\}^n \rightarrow \{0,1\},$$

$$f(x) = \begin{cases} 1 & \text{if } x = w \\ 0 & \text{else if,} \end{cases} \tag{3}$$

and

$$f_0(x) = \begin{cases} 0 & \text{if } x = 000...0 \\ 1 & \text{else,} \end{cases} \tag{4}$$

where the phase oracle is

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle, \tag{5}$$

where,

$$U_f : \begin{cases} |w\rangle \rightarrow -|w\rangle \\ |x\rangle \rightarrow |x\rangle & \forall x \neq w. \end{cases} \tag{6}$$

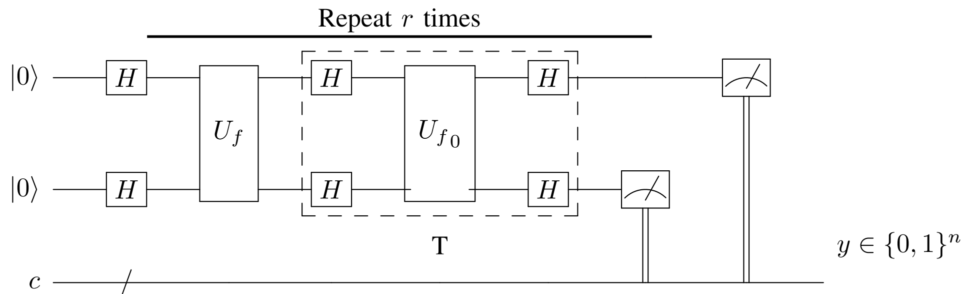

Fig. 1. Quantum circuit for the general version of Grover algorithm.

Then

$$U_f = 1 - 2\,|w\rangle\,\langle w|\,, \tag{7}$$

and

$$U_{f0} : \begin{cases} |0\rangle \to |0\rangle^{\otimes n} \\ |x\rangle \to -\,|x\rangle \quad \forall x \neq 00...000. \end{cases} \tag{8}$$

With this algorithm we want to find the input $x \in \{0,1\}^n$ such that $f(x) = 1$. With $f : \{0,1\}^n \to \{0,1\}$ as an unknown function, where we implemented U_f as an oracle. $y = w$ with highest probability. $T = H^{\otimes n} U_{f_0} H^{\otimes n}$, from eq. 8, we obtain $U_{f_0} = 2\big(\,|0\rangle\,\langle 0|\,\big)^{\otimes n} - I$. This result is known as reflection operator and it will be used soon [8].

Two registers used in the Grover's algorithm, n qubits in the first register and one qubit in the second register is the key architectural structure to achieve this speedup complexity over the classical algorithm [9]. We start the circuit for Grover's Algorithm by creating a superposition of 2^n computational basis states in the top register (general version of Grover's algorithm is shown fig. 1). All the qubits in the first register are initialized to state $|0, ..., 0\rangle$. After applying the n Hadamard gate, $H^{\otimes n}$, on the first we have the state:

$$|\psi\rangle = H^{\otimes n}|000...........0\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n}^{N-1} |x\rangle. \tag{9}$$

Note that, $|\psi\rangle$ is the superposition here. If we start the second register with a single qubit in state $|0\rangle$ or $|1\rangle$, after the Hadamard we achieve the respective Hadamard basis [2]. Let $f : \{0,1\}^N \to \{0,1\}$ be a function defined as:

$$f(x) = \begin{cases} 1 & \text{if x is the searched element} \\ 0 & \text{Otherwise.} \end{cases}$$

We define U_f as an oracle often referred to as a black box, defined as,

$$U_f(|i\rangle|j\rangle) = |i\rangle|j \otimes f(i)\rangle. \tag{10}$$

When we apply U_f on the state $|\psi\rangle$, the state of the second register does not changes [4] but the state of the first register changes, and we call this state $|\psi_1\rangle$. We assume that the Hadamard basis in the second qubit is $|-\rangle$. Here, $|\psi\rangle$ and $|\psi_1\rangle$ lives in $\mathcal{H}^N$. Equation 11 defines the effect of an oracle on the achieved superposition.

$$|\psi_1\rangle|-\rangle = U_f|\psi\rangle|-\rangle = \frac{1}{N}\sum_{i=0}^{N-1}(-1)^{f(i)}|i\rangle|-\rangle. \tag{11}$$

Due to *Quantum parallelism,*, we can observe all the database elements simultaneously at the quantum level. If the position of the searched element is known, then it will be labeled as the negative value of i in equation 11. It is impossible to get this result at the classical level. Before we perform the measurement and collapse our superposition in the classical bits, we apply another set of Hadamard gates, unitary operator, and n Hadamard gates for $O(\sqrt{N})$ times. Let us define this unitary operator, U_{f_0} as

$$T = 2|\psi\rangle\langle\psi| - I. \tag{12}$$

When we apply this operator on state ψ_1 we have,

$$\begin{aligned} |\psi\rangle &= \big(2|\psi\rangle\langle\psi| - I\big)|\psi_1\rangle \\ &= \sqrt{\frac{2^n - 1}{2^n}}\,|w^\dagger\rangle + \sqrt{\frac{1}{2^n}}\,|w\rangle\,. \end{aligned} \tag{13}$$

Equation (13) is the state of the first register and the second register is still on state $|-\rangle$ by assumption. Notice, $|\psi\rangle$ is in $\{w, w^\dagger\}$ states, w represents the summation of all states that can be solution to search problem; and $w^\dagger$ is the summation of all states which are no solutions. Coefficients are factor normalization constrained by the normalization condition, $\sum_i |c_i|^2 = 1$.

Thus, we can measure this state on both the register to evaluate the function f and get the probability finding the x record. Below, we present the implementation of this algorithm. Assuming we have an output of 1 for the function f with high probability, we calculate the probability of all possible input qubits on both the register. It is guaranteed to achieve the mentioned output.

II. Models, methods or materials

This section contains variational and kernel methods as potential models to implement Grover's algorithm as a quantum classifier with applications in Quantum Machine Learning. For this purpose, we expose those topics briefly. We shall commence with VQA, and at this end, we discuss some exciting areas of application.

A. Variational Quantum Algorithms

Variational Quantum Algorithms (VQA) address the circuit depth limit, and a limited number of qubits constrain current (near-term) quantum devices by implementing the classifier optimizer to train the parameterized quantum circuit. VQAs run the parameterized quantum circuit in the Quantum devices and the parameter optimization on the classical optimizer. VAQs mitigates the noise of the quantum circuit because it keeps the depth of the quantum circuit shallow. VQA is considered the prime proposal to achieve the quantum advantage with near-term quantum devices [10]. Given any problem (we consider classification for our simplicity), the first step is to define the loss (or cost) function C for our problem. C encodes the solution to our problem. We then perform the quantum operation called ansatz to optimize the parameter θ. The optimization task is defined as:

$$\theta^* = \arg\min_{\theta} C(\theta). \tag{14}$$

Equation (14) is trained in quantum-classical loop to obtain θ. One thing to note here is that while the classical optimizer is used to train θ, the VQAs use quantum devices to estimate C. This behavior is often considered the trade-off of VQAs. Once we define the cost function and ansatz, we are ready to train the parameter θ and solve the problem (14). Using the information in C gradient, it is proved that we can guarantee the speedup and convergence of optimizer for many optimization problems such as (14).

The most prominent implementation of VQA, sometimes also called *Quantum Neural Network (QNN)* is to tackle the classification task [11]. Here we briefly discuss the implementation of VQAs in the Grover search algorithm for classification. Du and Tao reformulated the classification task as the search algorithm using VQAs. The Grover-search-based quantum learning scheme (GBLS) dramatically reduces the number of measurements, and it outperformed the classical classifier in the measure of query complexity [9]. Following the optimization problem in (14), we can define the update rule for θ as:

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta}{B}\sum_{i=1}^{B} \nabla\mathcal{L}(\theta^{(t)}, B_i), \tag{15}$$

where η is the learning rate, B_i is the $i-th$ batch for *batch gradient descent* and B is the total number of batches. We can use varied B on GBLS for optimization of different quantum classifiers. In the classification process we use the grover-based searching only for the training and the prediction is done using optimized Variational Quantum Circuit (VQC). Recall from Grover 1996 article [5], the algorithm finds the record a from the dataset of size N by iteratively applying a predefined oracle

$$U_f = I - 2|a\rangle\langle a|, \tag{16}$$

and a diffusion operator defined on equation (12), and (9) as the input state. See figure 1 for the implementation of circuit for Grover algorithm. We are limited to change the oracle for this circuit. GBLS uses this property to define the classification problem as a searching problem. It replaces the oracle with VQC $U_{L_1} = \prod_{l=1}^{L} U(\theta^l)$ and multiple controlled qubits.

B. Kernels

In the previous section, we discussed how we could reformulate any classification task as a searching problem. In this section, we briefly discuss quantum kernels and quantum classifiers. By doing so, we would be able to introduce the concept of quantum classifiers in terms of quantum kernels and, hence, a searching problem that Grover's algorithm can solve.

The kernels method is an eminent tool in patterns analysis to identify non-linear relationships in any given dataset [12]. The fundamental kernel methods lie in data embedding into higher dimensional Hilbert space to be easy to analyze. The kernel method uses kernel functions that estimate the similarity between data in higher dimensional space by calculating their inner-product. We can switch between different models simply by switching between the kernels. In Quantum computing, this approach corresponds to change the data encoding strategy. Here we define a data encoding strategy. Let $\phi : \mathcal{X} \to \mathcal{F}$ be a feature map for a input space $\mathcal{X}$ and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ be a real or complex values positive definite functions for two data points.

Definition 1 (Modified from Def. 2 of [13])**.** *: Quantum Kernel is defined as the inner product between two data encoding feature vectors with $x, x' \in \mathcal{X}$*

$$k(x, x') = |\langle\phi(x')|\phi(x)\rangle|^2. \tag{17}$$

We define $\langle .|.\rangle$ as the inner product of two pure quantum states.

We can estimate eq. (17) using quantum computers that can calculate the inner product between two pure quantum states. Quantum models are often considered linear models in feature space.

Let us define a Hermitian operator$\mathcal{M}$ acting on a vectors in Hilbert space $\mathcal{H}$. We can define $\mathcal{M}$ as,

$$\mathcal{M} = \sum_i \alpha_i|\mu_i\rangle\langle\mu_i|, \tag{18}$$

where α_i are the eigenvalues of $\mathcal{M}$, $|.\rangle\langle.|$ is the outer product and $|\mu_i\rangle$ is an orthonormal basis in $\mathcal{H}$. Now we define quantum models as a function f of data input x,

$$f(x) = \langle\phi(x)|\mathcal{M}|\phi(x)\rangle. \tag{19}$$

Notice that eq.19 is in the form $\langle .|.\rangle$ and can be calculated as an inner product which we have defined as kernel methods. Thus, any quantum models can be considered kernel methods, and those models are a.k.a. quantum neural networks; however, based on the definition 1 those are closely related to kernel methods.

Ref. [12]–[15] investigates in-depth on quantum kernels. The scope of this paper is not to construct a quantum classifier but to relate quantum classifiers as kernel methods. The mathematical definition of VQA is closely related to the kernel methods.

In both of the approaches, the data are analyzed in higher-dimensional Hilbert space. Previously we discussed how VQAs could be used to reformulate classification tasks as searching problems. We can build an oracle on the Grover algorithm as per our desire and perform the classification.
Below, we present a simple implementation of building an oracle using the universal clauses only using AND, XOR, and OR gates. Our algorithm builds an oracle based on these gates such that it forces the output to be 1 before we supply it to the diffuser.

C. Methods

First, we need to get input information from the user. The user inputs the number of input qbits, number of clauses, and the clauses themselves. Our algorithm requires those information to be passed to construct an oracle. We call our algorithm universal because the user can input any combination of the clauses. The algorithm will output final probabilities based on the user inputs (evaluating the user clauses). The user is asked to input the clauses in a specific format. Consider the next example.

The user can input the following details:

- Number of qubits: 5.
- Number of clauses: 3.
- Clause 1: 0 AND 1.
- Clause 2: 1 XOR 2.
- Clause 3: 2 OR 3.

We have implemented XOR, OR, and AND gate, which we describe next.

Qbit indices are separated by the gate. So, the possible input clauses are 0 AND 1, 1 XOR 2, and 2 OR 3 in any order of the qubits. This is just one example. We can simulate this algorithm for any number of qubits and clauses (the number of clauses should be less than qubits), giving us the probability.

We implemented the XOR gate (Figure 2) using implemented using two CNOT gates. Our input qubits will be control variables, and they will output to one same qubit. Then, the output qubit will have value one if only one of the input qubits has value one.

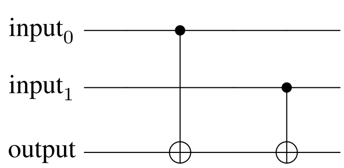

Fig. 2. XOR gate

The AND gate (fig. 3) is implemented using one Toffoli gate. If both input variables have value one, the output variable will also have value one. Otherwise, it will return zero.

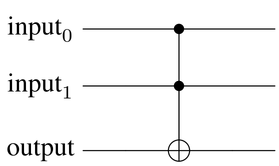

Fig. 3. AND gate

The OR gate (fig. 4) is implemented using one Toffoli and 2 CNOT gates. The output is one, if and only if one of the input gates has value 1. We performed all the coding implementation in the IBM Quantum framework [16].

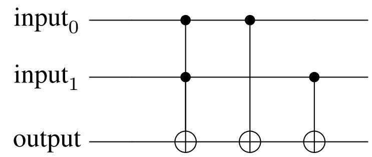

Fig. 4. OR gate

Figures 3-4 show the implementation of AND, XOR, and OR gates as described above.

For each clause gate, we use one clause qubit. Then, using Toffoli gate, we check if all of the clause qubits have value one, we return one. Otherwise, we return zero. We construct our oracle under this definition and the mathematical abstraction described in the introduction.

Before oracle, we place each input qubits into superposition by applying the Hadamard gate as described in the introduction. Thus, we have constructed an oracle. Finally, the diffuser is used $O(\sqrt{N})$ times before the measurement.

The diffuser is an essential part of Grover's algorithm. The diffuser function is universal. That means any two oracles that have the same number of input qbits can use the same diffuser. The function consists of the H-gates, X-gates, and multi-controlled Z-gate. We show an example for the circuit with 4 qubits and 3 universal clauses in Figure 5.

D. Algorithm and quantum circuit

This section contains information about the algorithm and the implemented circuit for our experiment. In the methods section, we described the process and steps to construct an oracle for the Grover algorithm using universal gates. Figure 5 shows an example of the circuit for the algorithm presented above. The circuit in the dotted box after the first set Hadamard gate is an oracle. We can define our oracle with clauses, gates and number of qubits as it is shown in the figure.

We also present an algorithm to construct user defined oracle in Algorithm 1. For more details about the algorithm please see the Methods section.

III. Results

In this section, we present two different examples. Figure 6 illustrates a histogram of probabilities when the input clauses are *0 AND 1* and *1 XOR 2* on 3 qubits circuit. 110 is the only answer, and it has a probability of 0.767. We explain this further. From our definition of the oracle, our oracle always outputs 1. So, qubit 0 and 1 outputs one if and only if both are 1, and 1 XOR 2 outputs one if and only if 2 is 0. 1 can not be 0 because it is one by clause 1. So, the only possible sequence of input to get one as an output of an oracle is 110. Our oracle, too, gives this answer.

Below we present another example with 4 qubits and 3 input clauses. Figure 7 shows a histogram. The Input clauses are *0 AND 3*, *1 XOR 2*, *2 AND 3*. If we follow the similar definition

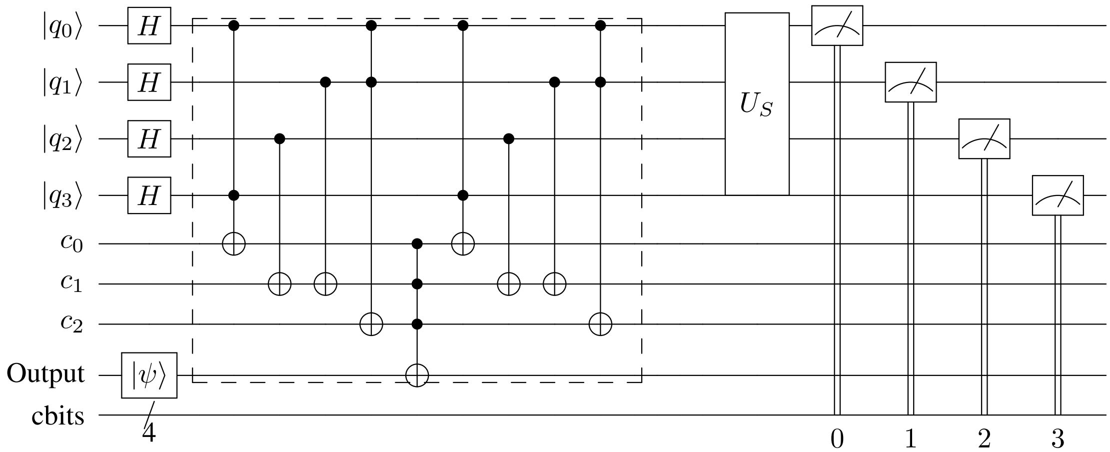

Fig. 5. Quantum circuit for Grover algorithm with the user defined clauses for an oracle. We used the clauses $|q_0\rangle$ AND $|q_3\rangle$, $|q_1\rangle$ XOR $|q_2\rangle$ and $|q_2\rangle$ AND $|q_3\rangle$. c_i refers to clauses, and cbits are the classical bits.

Algorithm 1 To construct a defined oracle.

Input: $q1, q2, ..., q_n$ and $c_1, ..., c_k$
Parameter: Set of universal clauses.
Output: Input bits that output 1.

1: for each input q:
2: $|\psi_1\rangle \longleftarrow$ Hadamard(q)
3: **for** each input clause c_i **do**
4: ORACLE:

$$\begin{cases} O_i & \longleftarrow c_i(q_j, q_k) \\ \text{out} & \longleftarrow \text{Toffoli}(O_1, O_2, ..., O_n) \end{cases}$$

5: **end for**
6: Grover(ORACLE)

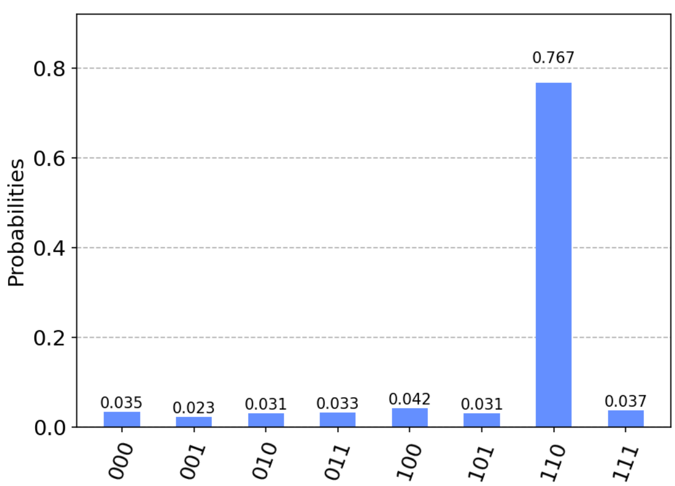

Fig. 6. 0 AND 1, 1 XOR 2

of obtaining 1 as in the above example we see that the only possible answer is 1011 and it has a probability of 0.473.

We see that the probability of the input sequences decreases as we increase the number of qubits. This is not always true.

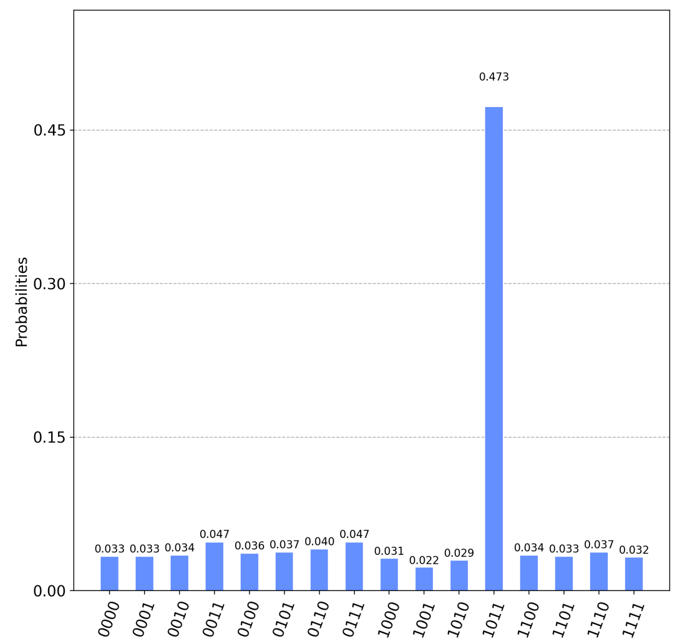

Fig. 7. 0 AND 3, 1 XOR 2, 2 AND 3

The only constrain it respect here is the sum of probability has to be 1. We will have multiple answers when we increase the number of qubits and the number of clauses. This is true because as we increase the number of qubits and clauses we will have different options to obtain 1 as an output. Also, we do not have any limitations on the number of clauses. The only condition on our algorithm is that the control qubit should come first, followed by universal gate followed by target qubit.

IV. Acknowledgements

The authors thank the Department of Computer Science at Baylor University for their support. This research was also funded, in part by the Baylor AI lab.

V. Conclusion

In this article, we discussed Grover's algorithm and its mathematical definition. We explained how variational quantum algorithms could be used in machine learning implementing Grover's algorithm. In particular, we explored an arena where we can reformulate the classification problem as a searching problem. We believe we can adopt variational quantum algorithms, e.g., for ML classification algorithms. This topic can be further explored and researched. We used the universal gates AND, XOR, and OR gates to construct an oracle for Grover's algorithm with universal clauses. Our constructed oracle finds a way to calculate the probabilities of possible input sequences such that the output of the function always yields 1. As we saw on Figures 6-7, the probability of the input qubit decreases with the increase in input qubits, which is the desired outcome.

As the next step, we will attempt to prove that we can reformulate any classification problem as a kernel method and vice versa. We will use this hypothesis to reformulate the classification problem as a searching problem. In the end, we will prove that we can solve this newly reformulated searching problem through Grover's algorithm with user-defined universal clauses oracle. Hence, we believe this research can be further expanded, exploiting Grover's algorithm when more working qubits are available along with access to powerful computers to simulate the process.

Acknowledgment

The work presented here was supported in part by the Baylor AI lab in Baylor University's Department of Computer Science.

References

[1] IBM, "Grover's algorithm," website, August 2021, https://qiskit.org/textbook/ch-algorithms/grover.html.

[2] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[3] G. Chen, S. A. Fulling, H. Lee, and M. O. Scully, "Grover's algorithm for multiobject search in quantum computing," in *Directions in Quantum Optics*. Springer, 2001, pp. 165–175.

[4] C. Lavor, L. Manssur, and R. Portugal, "Grover's algorithm: Quantum database search," *arXiv preprint quant-ph/0301079*, 2003.

[5] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–328, Jul 1997. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.325

[6] D. Deutsch, "Quantum theory, the church–turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[7] N. D. Mermin, "From cbits to qbits: Teaching computer scientists quantum mechanics," *American Journal of Physics*, vol. 71, no. 1, p. 23–30, Jan 2003. [Online]. Available: http://dx.doi.org/10.1119/1.1522741

[8] P. Rungta, "The quadratic speedup in grover's search algorithm from the entanglement perspective," *Physics Letters A*, vol. 373, no. 31, pp. 2652–2659, 2009.

[9] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, "A grover-search based quantum learning scheme for classification," *New Journal of Physics*, vol. 23, no. 2, p. 023020, feb 2021. [Online]. Available: https://doi.org/10.1088/1367-2630/abdefa

[10] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, pp. 1–20, 2021.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[12] D. K. Park, C. Blank, and F. Petruccione, "The theory of the quantum kernel-based binary classifier," *Physics Letters A*, vol. 384, no. 21, p. 126422, 2020.

[13] M. Schuld, "Supervised quantum machine learning models are kernel methods," *arXiv preprint arXiv:2101.11020*, 2021.

[14] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical review letters*, vol. 122, no. 4, p. 040504, 2019.

[15] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[16] IBM Q, 2021, Website: https://www.ibm.com/quantum-computing/.