



Universidade Federal Rural de Pernambuco

Departamento de Estatística e Informática

Programa de Pós-Graduação em Informática Aplicada

Algoritmo de Treinamento Não-Unitário Linear para Redes Neurais Quânticas Sem Peso

Ítalo Rodrigo Santos de Araujo

Recife

Julho de 2016

Ítalo Rodrigo Santos de Araujo

Algoritmo de Treinamento Não-Unitário Linear para Redes Neurais Quânticas Sem Peso

Orientador: Wilson Rosa de Oliveira Jr

Coorientador: Adenilton José da Silva

Dissertação de mestrado apresentada ao Curso de Pós-Graduação em Informática Aplicada da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Informática Aplicada.

Recife

Julho de 2016

À
minha mãe Maria Santos.

Agradecimentos

Aos meus pais, Maria do Carmo Santos de Araujo e Inaldo César de Araujo pelo apoio. Aos meus familiares, Irmão Inaldo César de Araujo Jr e Amigos Glaucia Tadu, José Accioly, Nielson Avelino, Lucas Sampaio, Alessanco Azevedo, Prof. Dr. Tiago Ferreira que contribuíram com bastante seriedade as noções de informática que precisei no início do mestrado. É um agradecimento especial a Ana Carolina Guimarães Fernandes que com muita paciência nunca deixou que eu desistisse desse grande desafio que foi realizar um mestrado em uma área que é diferente da minha formação que é Licenciatura Plena em Física.

Aos professores Dr. Wilson Rosa de Oliveira Jr e Dr. Adenilton José da Silva pela orientação, apoio e confiança que me deram para a realização desse trabalho de dissertação de mestrado e sempre tiveram paciência.

Agradeço imensamente a FACEPE/CNPq pelo apoio financeiro para a realização desse trabalho de dissertação de mestrado. E também a Universidade Federal Rural de Pernambuco pela realização do Programa de Pós-Graduação em Informática Aplicada (PPGIA).

Resumo

As Redes Neurais Quânticas [29] são formadas pela união de duas linhas de estudos a dada pela primeira linha as diretrizes das Redes Neurais Artificiais que foram propostas pelos pesquisadores Warren McCulloch e Walter Pitts, em 1943 [17], através da inspiração de um neurônio biológico e que realizava a classificação de estímulos nervosos a partir de padrões, sendo este processo nomeado de sinapses.

As Redes Neurais Artificiais ao serem um processo computacional algébrico que realizam a seleção de amostras mediante aprendizados que utilizam padrões pré-definidos ou não, a depender do tipo de treinamento da rede, proporcionando, desta forma, uma tomada de decisão.

Sendo a segunda linha, a Computação Quântica, realizada em 1982, pelo físico Richard Feynman [9], ao perceber a dificuldade de computadores convencionais em simularem sistemas puramente quânticos, propôs uma nova maneira de realizar computação ao partir de conceitos da mecânica quântica, a fim de realizar os circuitos e algoritmos quânticos [21] e esta nova maneira de realizar computação proporciona à mesma, através de estudos teóricos, características únicas como as superposições de estados quânticos, a acarretarem características probabilísticas aos estados quânticos, transformações unitárias para o processo de envio de informações, paralelismo quântico e aspecto de não clonagem [21].

Com a união dessas linhas de pesquisa que são das Redes Neurais Artificiais e a Computação Quântica, sendo a segunda, inserida ao conceito de qubits, que foi adicionado como elemento de formação da primeira, formaram-se as características quânticas. Tendo como objetivo de manipulação o envio de informações quânticas através de operadores, onde o conjunto dos mesmos descreve um circuito quântico lógico a formar assim as Redes Neurais

Quânticas Artificiais.

Neste trabalho irá utilizar uma Rede Neural Quântica Sem Peso do tipo Neurônio Lógico Probabilístico Quântico (q-PLN) [23] que a sua constituição é dada por um circuito quântico formado pelos estados e operadores quânticos e para a sua construção é de fácil implementação.

Nas literaturas [25] [26], existem diversos tipos de algoritmos de treinamento para Redes Neurais Quânticas que utilizam operações não-lineares. Esse trabalho propõe a utilização de um algoritmo que utiliza operadores lineares para o treinamento de Redes Neurais Quânticas Sem Peso do tipo q-PLN.

A solução apresentada por uma Rede Neural Quântica Artificial Sem Peso pode ser composta por uma quantidade de 2^n elementos em formato de superposição que pode apresentar resultados que satisfaçam ou não ao processo, já que utiliza processos de transformações não lineares. Essa quantidade de resultados propicia uma custosa verificação desses dados por uma máquina convencional.

Diante disso, esse trabalho apresenta uma alternativa que, além de permitir uma verificação menos custosa, permite a apresentação de resultados que satisfaçam o processo de análise de valores, sendo ela a utilização de operadores não-unitários em um processo de transformação linear para proporcionar resultados apenas compatíveis ao processo de verificação.

Neste trabalho, utiliza-se um algoritmo, representado por Algoritmo 1, que supõe a viabilidade da aplicação de operações lineares quânticas não-unitárias. Assim, a maneira de aprendizado que está sendo apresentada pelo Algoritmo 1 utiliza um circuito quântico composto por uma q-PLN com uma *qRAM* de tamanho qualquer.

Este algoritmo possui um custo de aprendizagem na ordem de $O(n_s * n_t)$, e proporciona a seleção de 100% dos acertos de todos os estados quânticos que se apresentem como possíveis soluções do problema.

Essa linha de pesquisa proporcionou o desenvolvimento de um novo método de aprendizado de uma Rede Neural Quântica Sem Peso do tipo p-PLN, por meio de operações lineares

que são procedimentos de fácil implementação e compreensão algébrica que é formada da manipulação de matrizes. Este novo método de aprendizado proporciona o enriquecimento do desenvolvimento referente ao estudo da rede neural citada anteriormente.

A linha de estudo deste trabalho pode proporcionar novas perspectivas com relação ao auxílio do estudo das Memórias Quânticas Associativas [34] como forma da descoberta de soluções. E como também, poderá ser possível a utilização deste algoritmo como auxílio da realização de buscas de valores em uma base de dados desordenada proporcionando resultados sendo assim associado ao algoritmo de Grover [12].

Palavras Chaves: Redes Neurais Quânticas Sem Peso, Operações Lineares, Operadores Não-unitários.

Abstract

The Quantum Neural Networks [29] are formed by the union of two study lines given by the first line, the guidelines of Artificial Neural Networks, that have been proposed by the researchers Warren McCulloch and Walter Pitts in 1943 [17] through the inspiration of a biological neuron, that performed the sort of nervous stimuli from standards. This process is named synapses.

The Artificial Neural Networks, being an algebraic computational process that performs the selection of samples by learning using predefined patterns or not, depending on the type of network training, providing thus a decision-making.

Being the second line, Quantum Computation, performed in 1982 by the physicist Richard Feynman [9], realizing the difficulty of conventional computers to simulate purely quantum systems, proposed a new way to perform computation starting from the mechanical quantum concepts, in order to accomplish the circuits and quantum algorithms [21]. This new way of performing computing provides to through theoretical studies, unique characteristics, such as superposition of quantum states, the igniting probabilistic characteristics of the quantum states; unitary transformations in the process of sending information, quantum parallelism, and non-cloning aspect [21].

With the union of these lines of research that are of Artificial Neural Networks and Quantum Computation, being the second one inserted into the concept of qubits, which was added as the first line element that formed the quantum characteristics. Aiming to handle quantum information sending by operators, where all of them describe a quantum logic circuit, thus forming the Artificial Quantum Neural Networks.

This work will use a Quantum Weightless Neural Networks of type Quantum Probabilistic

Logic Neuron (q-PLN) [23] that its constitution is given by a quantum circuit formed by the states and quantum operations and its construction is easy to implement.

In literatures [25] [26], there are several types of training algorithms for Quantum Neural Networks that use non-linear operations. This paper proposes the use of an algorithm that uses linear operators for training Quantum Weightless Neural Networks of type q-PLN.

The solution presented by a Quantum Weightless Neural Networks can consist of an amount of 2^n elements in superposition format, what can present results that gratify or not the process, since it uses non-linear transformation processes. This quantity of results provides a costly checking of the data in a conventional machine.

Therefore, this work presents an alternative that, in addition to allowing a less costly verification, allows the presentation of results that satisfy the process of values analysis. It must consist in the use of non-unitary operators in a linear transformation process to give results only compatible to the verification process.

In this work, an algorithm is used, represented by the Algorithm ??, which implies the feasibility of implementing non-linear quantum unit operations. So, the way of learning that is being presented by Algorithm 1 uses a quantum circuit composed by a q-PLN with a *qRAM* of any size.

This algorithm has a cost of learning in the order of $O(n_s * n_t)$, and provides a selection of 100% of hits of all quantum states that are regarded as possible solutions to the problem.

This line of research enabled the development of a new learning method for a quantum weightless neural networks of type q-PLN, through linear operations procedures that are easy to implement because they are formed by understanding algebraic manipulation matrices. This new learning method provides the enrichment of development related to the study of the neural network mentioned before.

The line of study of this work may provide new insights, in relation to the aid of the Associative Quantum Memory study [34] as a way of finding solutions. It also may be possible to use this algorithm to aid the realization of values searches in a disorderly database, providing results, thus being associated with Grover algorithm [12].

Key Words: Quantum Weightless Neural Networks, Linear Operations, Operators Non-unit.

Sumário

1	Introdução	1
1.1	Apresentação	1
1.2	Justificativa (motivação)	3
1.3	Objetivos	4
1.4	Contribuições obtidas	5
1.5	Organização do trabalho	5
2	Computação Quântica	7
2.1	Apresentação da Computação Quântica	7
2.2	Bit Quântico - Qubit	8
2.3	Muitos Qubits	9
2.4	Circuitos Quânticos	9
3	Redes Neurais Artificiais	15
3.1	Redes Neurais Artificiais	15
3.2	Redes Neurais Artificiais Com Peso	17
3.3	Redes Neurais Artificiais Sem Peso	19

3.4	Treinamento das Redes Neurais Artificiais Sem Peso	20
3.5	Redes Neurais Artificiais Quânticas Sem Peso	21
4	Algoritmo de Aprendizado Para o Treinamento de Redes Neurais Quânticas Sem Peso	27
4.1	Transformação Linear	27
4.1.1	Operadores Não-Unitários	28
4.2	Algoritmo de aprendizado	29
4.3	Exemplo1	33
4.3.1	Desenvolvimento algébrico do primeiro exemplo	33
4.4	Exemplo2	44
4.4.1	Desenvolvimento Algébrico do Segundo Exemplo	45
5	Conclusões e Trabalhos Futuros	52
5.1	Conclusão	52
5.2	Trabalhos Futuros	53
A	Álgebra Linear	58
A.1	Vetores	58
A.2	Bra $\langle\psi $	59
A.3	Combinação Linear	59
A.4	Produto Interno	60
A.5	Operadores Lineares	61
A.6	Produto Externo	61

A.7 Produto Tensorial	62
---------------------------------	----

Lista de Tabelas

- 2.1 Descrição da ação da porta lógica *CNOT*. 12
- 2.2 Descrição da ação da porta lógica *Toffoli*. 13

Lista de Figuras

2.1	Fio Quântico.	11
2.2	Porta lógica <i>NOT</i>	11
2.3	Porta lógica <i>Hadamard</i>	11
2.4	Porta lógica <i>CNOT</i>	12
2.5	Porta lógica <i>Toffoli</i>	12
2.6	Porta lógica <i>Toffoli</i>	13
3.1	Esquema de um neurônio biológico.	16
3.2	Gráfico linear de separação.	17
3.3	Nodo RAM.	19
3.4	Esquema de um Rede Neural Sem Peso.	20
3.5	qRAM1 [22].	22
3.6	Implementação da qRAM1 [22].	22
3.7	qRAM2 [22].	23
3.8	Implementação da qRAM2 [22].	23
3.9	Figura de uma Rede Neural Quântica Sem Peso Completa [22]	25
3.10	Figura de uma RNQSP.	26

4.1	Figura de uma Rede Neural Quântica Sem Peso completa.	34
4.2	Figura da implematação da MRNQSP	34
4.3	Operador1	36
4.4	Operador1	39
4.5	Operador1	41
4.6	Figura de uma RNQSP.	45
4.7	Figura da implementação da RNQSP	45

Capítulo 1

Introdução

A introdução inicia com a seção de Apresentação 1.1 que relata em poucas palavras o que é referente ao trabalho estudado. Na seção seguinte, a Justificativa 1.2 apresenta algumas vantagens de se desenvolver este trabalho. A seção seguinte, o Objetivo 1.3 indica as principais metas que serão desejadas em alcançar para o fechamento deste trabalho. Para a seção posterior, as Contribuições Obtidas 1.4 apresentam as contribuições que foram obtidos de resultados no final do estudo teórico. Na ultima seção, em Organização do Trabalho 1.5 é realizada uma breve apresentação de cada capítulo que será lido ao longo da dissertação.

1.1 Apresentação

Em 1980, Paul Benioff [3] ao perceber que informações poderiam ser enviadas mediante evolução de estados hamiltonianos, propôs, teoricamente, esse processo de envio representado por uma máquina de Turing Quântica. Em 1982, Richard Feynman [9] verificou que os computadores convencionais (ou clássicos) possuem dificuldade em realizar simulações de processos puramente quânticos e sugeriu que a construção de computadores que funcionassem por meio de elementos quânticos poderiam ser mais eficientes do que os computadores clássicos. A Computação Quântica proporciona uma nova maneira de realizar computação com características únicas, como a superposição de estados quânticos, emaranhamento; paralelismo quântico; e a não clonagem [21].

Uma das possíveis aplicações é da computação quântica é o desenvolvimento de modelos quânticos para a aprendizagem de máquina. Neste trabalho o foco será na iniciativa de unir a Computação Quântica com as Redes Neurais Artificiais (RNA) [13]. As RNA possuem a capacidade de aprender através de exemplos e são utilizadas em diversas aplicações práticas. Uma rede neural que utiliza os princípios da computação quântica é denominada uma Rede Neural Quântica [8].

As Redes Neurais Quânticas possuem o poder computacional de executar e analisar processos em paralelo por meio da propriedade do paralelismo quântico e manipulam estados quânticos em seu processamento. Se espera que as redes neurais quânticas possuam um maior poder computacional na resolução de tarefas.

As Redes Neurais Sem Peso foram inicialmente propostas em [1] e possuem diversas aplicações práticas [6]. Em [5] é desenvolvido um modelo de rede neural quântica sem pesos. As redes neurais quânticas sem pesos apresentam em sua formação um circuito lógico arquitetado por operadores e estados quânticos, sendo de fácil implementação em hardware. O processamento de uma rede neural quântica sem pesos utiliza apenas transformações unitárias.

A busca por um modelo de rede neural quântica e seu algoritmo de aprendizado ainda é um problema em aberto [29]. Em diversos trabalhos são utilizadas operações não lineares como forma de seleção executada por uma rede neural.

Esse trabalho tem como objetivo apresentar um algoritmo de aprendizado para redes neurais quânticas que utiliza apenas transformação não-unitário linear. Esse algoritmo proposto é baseado no princípio da superposição, utiliza operadores não unitários e possui custo computacional em $O(n_s \cdot n_t)$ no qual n_s é representado pelo número de *Seletores* da rede neural e n_t corresponde a quantidade de *Padrões* que serão analisados em todo o processo do algoritmo.

1.2 Justificativa (motivação)

As Redes Neurais Artificiais (RNAs) possuem uma infinidade de aplicações como na área da engenharia, ciências exatas e como também na biologia. Essas redes foram formadas a partir da inspiração das redes neurais biológicas. As RNAs possuem um poder de processamento limitado com relação às redes neurais biológicas por terem limitações em não processar uma grande quantidade de informações ao mesmo tempo e também por não proporcionarem uma elevada velocidade no envio de informação com relação às redes neurais biológicas [20].

Para as RNAs tentarem corresponder ao desempenho das redes neurais biológicas a mesma teria que manipular uma grande quantidade de energia, sendo assim, esta grandeza física pode proporcionar temperaturas elevadas nos componentes elétricos acarretando o efeito joule que pode proporcionar o derretimento dos equipamentos por conta da alta temperatura.

Para o processamento de uma massiva base de dados grandes empresas, como Google, Microsoft e NASA, necessitam de ferramentas computacionais que tenham um maior poder de execução de tarefas. Uma iniciativa encontrada foi o investimento na Computação Quântica.

A Computação Quântica é formada pela manipulação de estados quânticos por meio de operadores quânticos. Tudo isso sendo realizado em um sistema controlado, fora de interferências externas para não ocorrer uma interferência sobre o resultado final. Esta computação possui a propriedade em realizar processos reversíveis ou não dissipativos. Essa característica indica que a energia que proporcionou uma informação de entrada ϕ_0 terá uma saída de informação ϕ_1 com a mesma quantidade de energia que foi manuseada inicialmente [10] [10] [33].

Pelo conhecimento das limitações das RNAs houve a iniciativa de unir a com a área da Computação Quântica afim de potencializar o desempenho dessas redes. Desta forma surgiram as Redes Neurais Quânticas Artificiais que realizam em sua manipulação de informação o qubit que representa a menor unidade de informação utilizada por um computador quântico. A manipulação dessa informação é tratada por operadores lógicos quânticos que são reversíveis e unitários. As redes neurais quânticas possuem um poder maior de

aprendizado por manipular uma quantidade de informação utilizando processos lineares e ou não-lineares [25].

Já existe na literatura diversos trabalhos sobre redes neurais quânticas. O foco deste trabalho será no modelos de redes neurais quânticas sem pesos, pelos seguintes critérios:

- Facil implementação em hardware.
- Possuir alta velocidade de execução [24].
- Não interferir no processamento do hardware, de sorte que os resultados dessas redes são proporcionados por uma manipulação algébrica de matrizes fixas [24].

As Redes Neurais Quânticas Sem Pesos são compostas por operadores unitários em sua constituição e este trabalho propõe um algoritmo de aprendizado que utiliza as transformações lineares com operadores não-unitários [14] [11] como limitante final do processo de seleção dos estados quânticos em superposições que permitam as mais prováveis soluções para o problema tratado.

1.3 Objetivos

O objetivo deste trabalho é definir um algoritmo que realize o aprendizado de uma rede neural quântica utilizando operações lineares e operadores não-unitários associados as Redes. Os objetivos específicos deste trabalho são descritos a seguir:

1. Realizar uma revisão de literatura para identificar os problemas em aberto na área de redes neurais quânticas. Como a utilização de operações lineares para treinamento de Redes Neurais Quânticas Sem Peso.
2. Propor um algoritmo de treinamento para redes neurais quânticas sem peso que utilize apenas transformações não-unitário lineares.
3. Verificar o algoritmo por métodos algébricos a sua validade.

1.4 Contribuições obtidas

As contribuições obtidas neste trabalho são descritas nos tópicos a seguir:

1. Na revisão da literatura realizada nesse trabalho não foi possível encontrar estudos sobre Redes Neurais Quantidade Sem Peso que utilizassem apenas transformações lineares em seu algoritmo de aprendizado.
2. Foi proposto um algoritmo para o treinamento de redes neurais quânticas sem pesos que utilizasse apenas transformações lineares, com operadores não-unitários.
3. O processamento do algoritmo foi validado com execuções algébricas em pequenos exemplos que permitiram mostrar a viabilidade do algoritmo proposto.
4. O algoritmo possui um custo computacional de execução de $O(n_t \cdot n_s)$ em que n_t representa o número de *Padrões* a serem analisados e n_s representa a quantidade de *Seletores* que é utilizada em todo o processo.
5. Esse algoritmo é uma maneira simples de construir e de realizar um processo de aprendizagem de circuitos quânticos que representam uma Rede Neural Quântica Sem Peso de formato genérico.

1.5 Organização do trabalho

O restante deste trabalho está organizado da seguinte forma. O capítulo 2 apresenta uma breve introdução dos elementos da Computação Quântica que serão utilizados para o estudo e análise do problema. As manipulações algébricas desse capítulo são referenciadas no Apêndice A que apresenta alguns elementos da álgebra linear.

No capítulo 3 é mostrado, em poucas palavras, um panorama do que seja Rede Neural Artificial, além de apresentar a estrutura de uma Rede Neural Quântica Sem Peso que será utilizada como ferramenta para a realização deste trabalho de dissertação. O capítulo ?? realiza a apresentação dos operadores não-unitários com suas referências e aplicações na problemática proposta desse trabalho.

No capítulo 4 é apresentado um algoritmo de aprendizagem para redes neurais quânticas sem pesos que utiliza apenas transformações lineares. Este capítulo também apresenta exemplos de execução do algoritmo proposto. Por fim, a conclusão apresenta os resultados obtidos nesse trabalho, além dos projetos futuros.

Capítulo 2

Computação Quântica

Neste capítulo são apresentados os elementos da Computação Quântica que serão utilizados para a realização deste trabalho. Na seção 2.1, será feita uma breve apresentação da Computação Quântica. Em seguida, na seção 2.2, será apresentada a menor unidade de informação que é utilizada em um computador quântico, o qubit, e como ele é representado de maneira algébrica. Posteriormente, na seção 2.3, mostraremos como representar e manipular muitos qubits. Já na seção 2.4 apresentaremos os elementos básicos de um circuito quântico.

2.1 Apresentação da Computação Quântica

Exposto por Paul Benioff em 1980, os primeiros passos da Computação Quântica foram iniciados a partir de estudos que indicaram a formação de processos quânticos computacionais que funcionavam através da evolução de estados hamiltonianos, os quais deram origem às primeiras teorias sobre computadores quânticos. Estes processos quânticos foram propostos pela a execução de uma máquina de Turing Quântica que foi proposta por Benioff.

Posteriormente, o físico Richard Feynman, em 1982 [9], ao notar a dificuldade de computadores convencionais simularem sistemas puramente quânticos, sugeriu uma nova maneira de realizar computação partindo de conceitos da mecânica quântica. Em [7] é apresentado o primeiro algoritmo puramente quântico que foi desenvolvido com a propriedade de possuir a

característica da superposição de estados e do paralelismo quântico [21]. Com o passar dos anos surgiram outros algoritmos como o de Grover [12] que realiza a busca em uma base de dados desordenada com custo na ordem de $O(\sqrt{N})$, sendo N o número total de elementos da lista, e o algoritmo de Shor's [30] que motivou o prosseguimento do estudo da Computação Quântica, valendo ressaltar que o algoritmo de Shor realiza um processo de fatoração com maior rapidez que os algoritmos clássicos até então existentes.

2.2 Bit Quântico - Qubit

Tido como a menor unidade de informação utilizada em um computador quântico, o qubit [21], também denominado de bit quântico e aplicado para manipulação de informações na Computação Quântica. O conceito de qubit é análogo ao conceito do bit clássico, menor unidade de informação usada nos computadores convencionais (nomeados de computadores clássicos), a qual possui dois possíveis resultados de estados de informação, 0 ou 1 o que permite sua interpretação através de circuitos elétricos simples onde o estado 0 não possui corrente elétrica, ao passo que o estado 1 indica a presença de corrente elétrica caracterizando, assim, a lógica booleana (lógica binária). O bit quântico, por sua vez, pode ser manipulado, teoricamente, com o conhecimento da álgebra linear, comentado no apêndice A, no ambiente do espaço vetorial de Hilbert, representado pela notação de Dirac $|\cdot\rangle$.

Um qubit é descrito pela Equação 2.1

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (2.1)$$

onde os valores de α e β são chamados de *amplitudes* e ambos são números complexos onde $|\alpha|^2$ representa a probabilidade de ocorrer o estado puro $|0\rangle$ e $|\beta|^2$ indica a probabilidade de ocorrer o estado puro $|1\rangle$. Essas duas informações correspondem a relação de $|\beta|^2 + |\alpha|^2 = 1$. E $|0\rangle$ e o $|1\rangle$ são descritos na Equação 2.2.

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2.2)$$

Estes vetores são ortonormais, o transposto conjugado de um qubit $|\psi\rangle$ é denotado por $\langle\psi|$, a equação 2.3 apresenta o transposto conjugado dos qubits $|0\rangle$ e $|1\rangle$. A equação 2.4 apresenta o *Produto Interno* dos vetores $|0\rangle$ e $|1\rangle$.

$$\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad \langle 1| = \begin{pmatrix} 0 & 1 \end{pmatrix} \quad (2.3)$$

$$\begin{aligned} \langle 0|0\rangle &= \langle 1|1\rangle = 1 \\ \langle 1|0\rangle &= \langle 1|0\rangle = 0 \end{aligned} \quad (2.4)$$

2.3 Muitos Qubits

A manipulação de muitos qubits é exercida com a ferramenta do *Produto Tensorial* [27] \otimes , comentada no apêndice A, representado pelo vetor $|\gamma\rangle$, o qual é composto dos estados $|\psi\rangle$ e $|\varphi\rangle$ que podem ser representados por:

$$|\gamma\rangle = |\psi\rangle |\varphi\rangle = |\psi\rangle \otimes |\varphi\rangle = |\psi\varphi\rangle \quad (2.5)$$

$$|\gamma\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} \otimes \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} = \begin{pmatrix} \psi_1 \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} \\ \psi_2 \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \psi_1\varphi_1 \\ \psi_1\varphi_2 \\ \psi_2\varphi_1 \\ \psi_2\varphi_2 \end{pmatrix} \quad (2.6)$$

2.4 Circuitos Quânticos

Os computadores convencionais são constituídos por circuitos clássicos e portas lógicas clássicas como ferramentas para o manuseio e manipulação dos bits em qualquer processa-

mento de informação em seus componentes elétricos [21]. Esses elementos lógicos são tidos como irreversíveis [21] por não ser possível saber com precisão qual seria a codificação de entrada para uma dada saída correspondente.

O circuito quântico é formado pelo agrupamento de portas lógicas quânticas, também chamadas de operadores quânticos, com o objetivo de manipular estados quânticos formados por qubits. Estes operadores são reversíveis, pois é possível saber com segurança qual ou quais entradas foram utilizadas para corresponder a um determinado resultado de saída de um circuito quântico característico.

Os operadores quânticos são unitários [21] e podem ser representados por meio da álgebra linear, apêndice A, que é composta pelo formato matricial de $2^n \times 2^n$, onde n representa o número de qubits que uma dada porta lógica manipula em seu funcionamento. Para demonstrar o processo unitário dos operadores quânticos, este trabalho utilizou a manipulação da porta lógica quântica *Hadamard* (H) [21], apresentada na base computacional (canônica) matricial, dada por:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.7)$$

O procedimento de manipulação para a entrada de um operador é composto pela utilização de um qubit de estado $|a\rangle$, o qual possui a mesma dimensionalidade do operador. A seguir, é demonstrada, de maneira algébrica, a atuação deste operador com um qubit no estado $|0\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (2.8)$$

Os operadores quânticos manipulam uma quantidade de qubits fixa. Sendo assim, para ser possível a manipulação de qualquer quantidade de estado é necessária a execução da ferramenta do *Produto Tensorial* [21], descrita no apêndice A.

A propagação de informações acerca da Computação Quântica é representada por uma espécie de fios quânticos, representados por uma reta que envia a informação da direita para a esquerda, mostrada na Figura 2.4. Nas conexões desses fios quânticos podem existir uma infinidade de portas lógicas quânticas com entradas e saídas que correspondem a inúmeros estados de informação possivelmente distintos.

Assim, é de fundamental importância para este trabalho os seguintes operadores *NOT* [21], *Hadamard* [21] e *CNOT* [21], pois os dois primeiros utilizam um qubit, enquanto o último funciona utilizando dois qubits, sendo um de controle, representado graficamente por um ponto preto (\bullet) Figura 2.4, e o outro de alvo que corresponde graficamente ao outro qubit em \oplus .



Figura 2.1: Fio Quântico.

Em seguida, são representados os operadores em formato gráfico e as entradas e saídas correspondentes em notação de Dirac para a entrada dos estados $|0\rangle$ e $|1\rangle$. Já os resultados da porta *CNOT* são representados na Tabela 2.4.

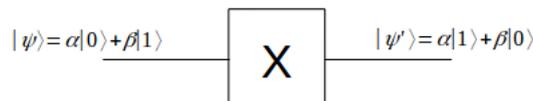


Figura 2.2: Porta lógica *NOT*.

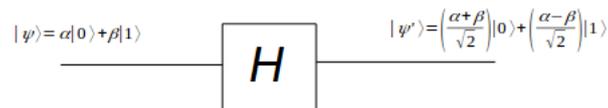
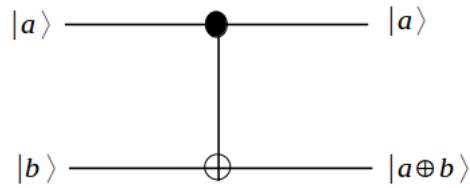
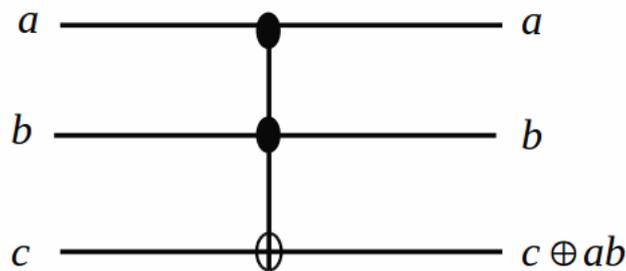


Figura 2.3: Porta lógica *Hadamard*.

Por meio de algumas portas lógicas anteriores é possível formar o operador *Toffoli* [21] que possui três qubits, sendo dois de controle e um de alvo. O seu funcionamento é semelhante ao da porta *CNOT*. A representação gráfica de *Toffoli* é dada na Figura 2.4.

Figura 2.4: Porta lógica *CNOT*.

Entrada		Saída	
a	b	a'	b'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

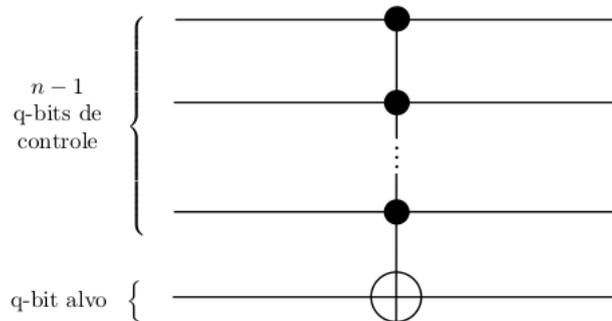
Tabela 2.1: Descrição da ação da porta lógica *CNOT*.Figura 2.5: Porta lógica *Toffoli*.

E a sua tabela verdade é apresentada pela Tabela 2.4.

Também é possível representar uma porta *Toffoli* genérica que possui n qubits de controle para um único alvo pela Figura 2.4.

A representação matricial com a dimensionalidade utilizada neste trabalho pelos opera-

Entrada			Saída		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	1	1
1	1	1	1	1	0

Tabela 2.2: Descrição da ação da porta lógica *Toffoli*.Figura 2.6: Porta lógica *Toffoli*.

dores citados anteriormente são descritos a seguir:

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (2.9)$$

$$Hadamard = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.10)$$

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.11)$$

Capítulo 3

Redes Neurais Artificiais

Neste capítulo será apresentado na seção 3.1 um panorama explicativo do que seja uma Rede Neural Artificial, algumas indicações de sua aplicação, alguns exemplos de classificação de aprendizagem e apresentação de sua arquitetura. Em sequência, na seção 3.2, será apresentado um tipo de Rede Neural Artificial que utiliza peso em seu critério de aprendizagem. Posteriormente, a seção 3.3 será composta da apresentação geral de uma rede neural sem peso e sua arquitetura básica. Já na seção 3.4 será demonstrado um tipo de treinamento de uma Rede Neural Artificial Sem Peso utilizada por meio de discriminadores. Por fim, na seção 3.5 será apresentada a arquitetura de uma Rede Neural Quântica Sem Peso que foi utilizada como ferramenta para a manipulação dos resultados deste trabalho de dissertação.

3.1 Redes Neurais Artificiais

Propostas pelos pesquisadores Warren McCulloch e Walter Pitts, em 1943 [17], através da inspiração de um neurônio biológico o qual pode ser representado pela Figura 3.1 e que realizava a classificação de estímulos nervosos a partir de padrões, sendo este processo nomeado de sinapses, as Redes Neurais Artificiais são processos computacionais algébricos que realizam a seleção de amostras mediante aprendizados que utilizam padrões pré-definidos ou não, a depender do tipo de treinamento da rede, proporcionando, desta forma, uma tomada de decisão.

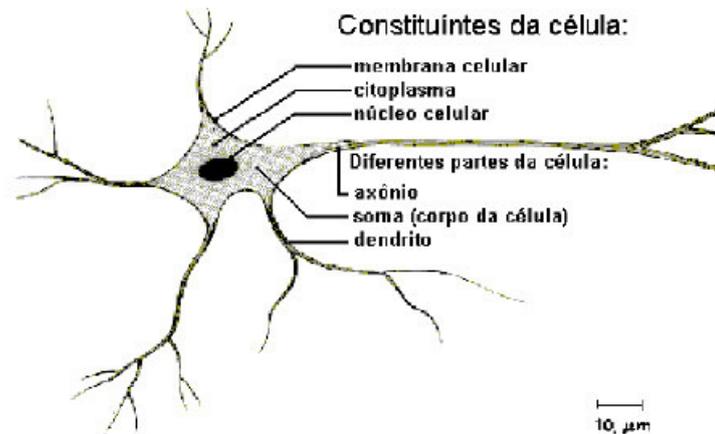


Figura 3.1: Esquema de um neurônio biológico.

Pode-se verificar a aplicabilidade dessas redes em variadas áreas do conhecimento como na computação evolucionária [18], na área de saúde para a análise de exames a exemplo da verificação do câncer por meio de mamografias [19], na manipulação de games para indicar a melhor forma de realizar uma tomada de decisão, na robótica com o objetivo de criar máquinas mais independentes [16], etc.

As redes neurais artificiais, enquanto processos computacionais algébricos, selecionam amostras a partir de diversas espécies de aprendizados diferentes, conforme demonstrado abaixo, sendo esses métodos apresentados pela referência [13]

- Aprendizado por correção de erro;
- Aprendizado Baseado em memória;
- Aprendizado Hebbiana;
- Aprendizado Competitiva;
- Aprendizado Boltzmann;
- Aprendizado com um Professor;
- Aprendizado sem Professor.

Essas redes possuem, em sua arquitetura, três partes distintas a **Entrada**, o **Nível Intermediário** e a **Saída**. A **Entrada**, início do processo, é composta por i elementos, onde

$i \in \mathbb{N}$, responsáveis pelo recebimento das amostras a serem analisadas pela rede para posterior envio a próxima parte que é o **Nível Intermediário**. Este, por sua vez, corresponde ao local de processamento dos dados propriamente ditos, realizado por uma técnica específica da Rede Neural Artificial. Por fim, os resultados serão apresentados à **Saída**.

As Redes Neurais Artificiais podem ser classificadas em redes com ou sem peso. As redes neurais artificiais com peso apresentam diversas espécies, uma delas foi proposta por Rosenblatt [28], nomeada de Perceptron, que realiza a classificação de padrões por meio de ajuste dos pesos para a solução de problemas linearmente separáveis [13]. Já a outra forma, de sorte que utilizaremos neste trabalho, **não** possui o peso como critério de triagem dos resultados, sendo assim nomeada de Rede Neural Artificial Sem Peso, que será tratada no tópico 3.3.

3.2 Redes Neurais Artificiais Com Peso

Consoante afirmado acima, a rede neural artificial com peso pode ser evidenciada pela Rede Neural Artificial Perceptron, proposta primeiramente por Rosenblatt em 1958 [28], que resolve problemas linearmente separáveis, quando de camada única, como mostrado na Figura 3.2, onde a seleção das amostras pode ser feita mediante uma reta com uma separação.

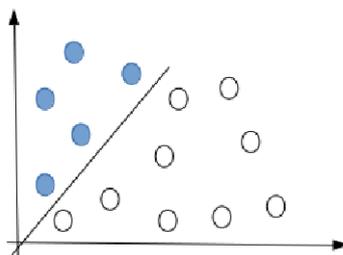


Figura 3.2: Gráfico linear de separação.

Abaixo, segue a demonstração do funcionamento de um Perceptron, a partir de uma demonstração algébrica, em que temos um neurônio que executa a adição de amostras nas entradas de tamanho finito x_k , onde essas amostras possuem a dimensionalidade de um vetor a ser manipulado pelo método do produto interno do vetor dos pesos $w_k = [w_1, w_2, \dots, w_k]$ correspondente as suas respectivas amostras, sendo k um número inteiro.

Essa demonstração é indicada pela Equação (3.1), sendo m a quantidade de realizações dos processos do produto interno e t a quantidade realizada de interação pelo perceptron que pode ser encarado como o tempo.

$$y(t) = \sum_{k=1}^m x_k(t) * w_k(y) \quad (3.1)$$

O aprendizado desta rede é composto pela correção dos *Erros*, representado por e , pelos derivadas da *Saída*, representados por y , e pela *Saída Desejada*, representada por d , que juntos correspondem a Equação (3.2):

$$e(t) = d(y) - y(t) \quad (3.2)$$

Na equação acima referida, quanto menor for o *Erro* melhor será o processamento da rede perceptron, sendo o aprendizado realizado por meio dos ajustes dos pesos mediante alguma heurística de adaptação para a realização do *Teorema da Convergência* [13], o qual, por sua vez, faz com que os resultados da rede neural tendam para um valor desejado. Assim, os processos de ajustes dos pesos podem ser descritos pela equação (3.3):

$$w(t + 1) = w(t) + \eta e x(t) \quad (3.3)$$

Na equação acima, η é a constante de aprendizado que irá proporcionar a oscilação dos valores de *Saída*, representado por y . Dessa forma, um alto valor de η corresponderá a resultados y 's que podem se aproximar ou se afastar desmedidamente, por poucas cotações de tempo, representado por t , da *Saída Desejada*, representada por d .

Ao contrário, se η for baixo, então y 's terá aproximações ou afastamentos sucintos com relação a d com o decorrer do aumento de t .

A classificação dos resultados é dada pela comparação com um valor limiar θ , que segue a restrição abaixo, de modo que y maior que θ corresponderá ao padrão. Caso contrário, y não será definido como solução e será descartado.

$$y = \begin{cases} 1 & \text{if } \sum w_k x_i \geq \theta \\ 0 & \text{if } \sum w_k x_i < \theta \end{cases} \quad (3.4)$$

3.3 Redes Neurais Artificiais Sem Peso

Reconhecidas por analisar amostras mediante alocação de memória, as Redes Neurais Artificiais Sem Peso são compostas por tabelas verdades com valores booleanos 0's ou 1's que se localizam em um ponto da rede nomeado de *RAM* (*Random Access Memory*), mostrado na Figura 3.3, que realiza a comparação dos dados a serem analisados por indicação de cada elemento de entrada para um nodo correspondente da *RAM*, o que proporciona a indicação de uma resposta que resulte em uma solução que terá uma maior frequência.

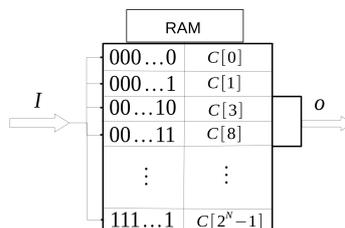


Figura 3.3: Nodo RAM.

As Redes Neurais Artificiais Sem Peso podem ser compostas por uma quantidade qualquer de *RAM*'s representada na Figura 3.4.

Como na literatura são encontradas muitas formas de Rede Neural Artificial Sem Peso e aplicações correspondentes, utilizaremos neste trabalho a rede do tipo *pRAM*, que aceita e manipula os valores 0's e 1's como informação e os valores entre 0 e 1 como resultados de saída sendo, desta forma, caracterizada como uma rede probabilística.

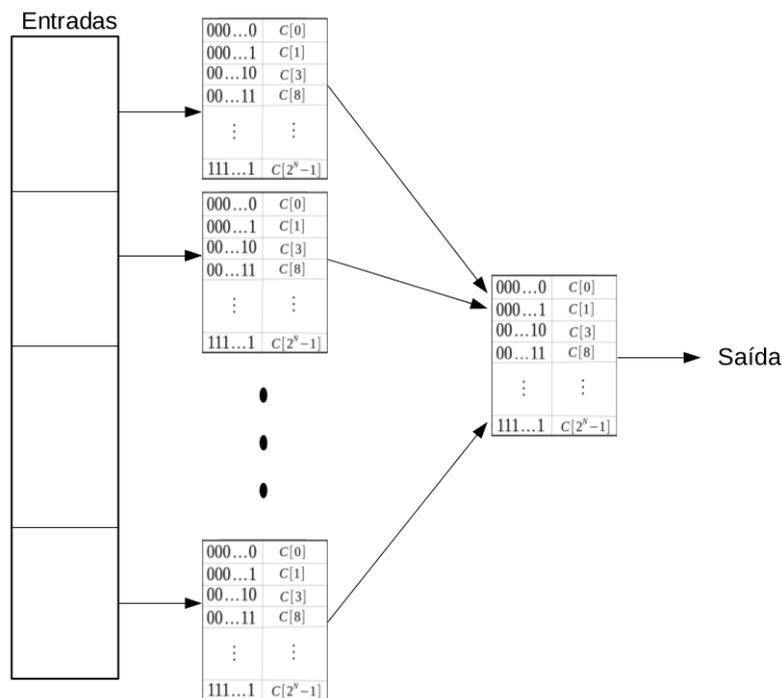


Figura 3.4: Esquema de um Rede Neural Sem Peso.

3.4 Treinamento das Redes Neurais Artificiais Sem Peso

Uma Rede Neural Artificial Sem Peso, de uma única camada, possui um *discriminador* que é constituído por K camadas de *RAM* com N nodos. Esses *discriminadores* podem receber a quantidade KN de bits de informação, possuindo poder para analisar um tipo de padrão.

O início desse processo de aprendizagem é composto por todas as posições de memórias iguais a 0. O seu treinamento é substituir determinados valores de 0's que estão alocados em algum endereço de memória por algum valor padrão p . Após esse processo, será obtida uma quantidade total de p , de sorte que o *discriminador* estará apto a identificar o padrão que foi treinando.

Para a Rede Neural Sem Peso realizar a verificação de n padrões esta tem que possuir uma quantidade n de *discriminadores* acoplados de maneira paralela no sistema. Sendo assim, essa quantidade de *multidiscriminadores* é capaz de realizar o método nomeado de Wisard.

A técnica Wisard é composta por uma quantidade n de *discriminadores* que analisam uma quantidade corresponde de n padrões que podem ser apresentados como imagens ou valores de bits.

Para uma rede neural com *multidiscriminadores* o resultado pode ser apresentado em valores binários 0, 1 ou uma porcentagem de acertos das apresentações das análises.

Como mencionado anteriormente, os resultados das *pRAMs* são nodos de redes neurais que compõem os *Neurônios Lógicos Probabilísticos*.

Estes nodos apresentam os resultados dos seguintes estados de *Saída*, representado por y , onde x representa um valor qualquer de entrada e $\sigma[\cdot]$ o processamento da *pRAM*.

$$y = \begin{cases} 0 & \text{se } \sigma[x] = 0 \\ 1 & \text{se } \sigma[x] = 1 \\ \text{random}(0,1) & \text{se } \sigma[x] = u \end{cases} \quad (3.5)$$

O resultado 1 indica a análise correta das entradas. Já o resultado 0 representa o erro, ao passo que o resultado u indica um valor entre 0 e 1 que pode ser interpretado como a probabilidade da aproximação do acerto ou do erro.

3.5 Redes Neurais Artificiais Quânticas Sem Peso

As Redes Neurais Artificiais Quânticas [29] são formadas a partir das Redes Neurais Artificiais e de elementos da Computação Quântica utilizando, para isso, do conceito de qubits para a manipulação e o envio de informação quântica através de operadores quânticos, onde o conjunto desses elementos quânticos descrevem um circuito quântico.

Assim, a partir do conhecimento deste tipo de estrutura, ou seja das Redes Neurais Artificiais Quânticas, houve a necessidade de realizar uma Rede Neural Artificial Quântica Sem Peso [31] utilizando a analogia das redes Neurais Artificiais Sem Peso com a característica

do *Neurônio Lógico Probabilístico*(PLN) e com o nodo da *pRAM*. No qual, este neurônio proporcionou a realização do *Neurônio Lógico Probabilístico Quântico*(q-PLN) [23] que esse possui uma *qRAM*.

Esse tipo de Rede Neural Quântica Sem Peso utiliza a *qRAM* que equivale ao conceito análogo quântico da *pRAM* que é clássica, ao passo que a *qRAM* é formada por elementos da Computação Quântica que são nomeados de operadores quânticos e que proporcionam os resultados finais em formato probabilístico de estados.

Neste trabalho foram utilizadas duas formas de *qRAMs*, uma que possui uma entrada e outra que recebe duas entradas, mas ambas foram formadas graficamente pela ferramenta *Quantum* do software Mathematica [22].

A primeira *qRAM*, denominada de *qRAM1*, está descrita na Figura 3.5, ao passo que a segunda na referência [31], denominada neste trabalho de *qRAM2*, apresentada na Figura 3.7. A implementação da Figura 3.5 e Figura 3.7 realizada por meio da ferramenta *Quantum* do software Mathematica [22] está descrita, respectivamente, na Figura 3.6 e na Figura 3.8.

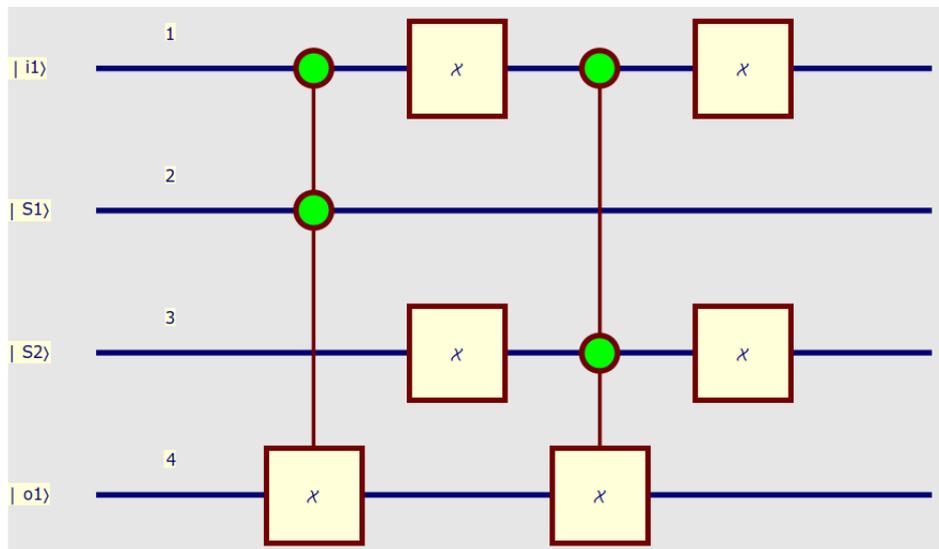


Figura 3.5: qRAM1 [22].

$$\text{QuantumPlot}[x_3 \otimes x_1 \cdot c^{\{i,3\}} [x_4] \cdot x_3 \otimes x_1 \cdot c^{\{i,2\}} [x_4]]$$

Figura 3.6: Implementação da qRAM1 [22].

A representação da palavra [QuantumPlot](#) nas Figura 3.8 e Figura 3.6 fazem referência a

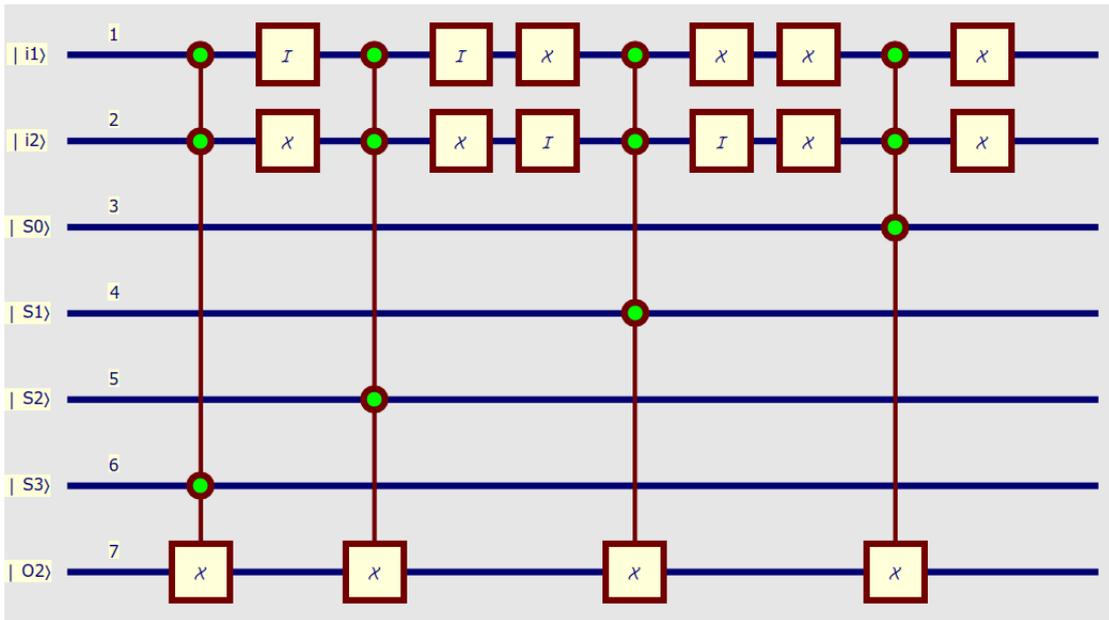


Figura 3.7: qRAM2 [22].

$$\text{QuantumPlot}[(x_2 \otimes x_1 \cdot c^{[i, \hat{2}, \hat{3}]} [x_7] \cdot x_1 \otimes x_2) \cdot (x_2 \otimes x_1 \cdot c^{[i, \hat{2}, \hat{4}]} [x_7] \cdot x_1 \otimes x_2) \cdot (x_1 \otimes x_2 \cdot c^{[i, \hat{2}, \hat{5}]} [x_7] \cdot x_2 \otimes x_1) \cdot (c^{[i, \hat{2}, \hat{6}]} [x_7])]$$

Figura 3.8: Implementação da qRAM2 [22].

uma função do software Mathematica que transforma as expressões escritas em sua entrada em formato dos circuitos quânticos.

A *qRAM1* é composta pelos seguintes componentes: uma *Entrada*, dois *Seletores* e uma *Saída*. O seu estado inicial $ket\psi_{0_1,0_2,0_3,0_4}$ será manipulado pela *qRAM1* em ordem de funcionamento a partir dos seguintes passos regidos pelas Equações 3.6 3.7, abaixo demonstrada, para resultar uma *Saída* dada pela quarta linha do circuito quântico, demonstrado na Figura 3.5.

$$|\psi_1\rangle = (2XOR_{i_1s_1o}) |\psi_0\rangle \tag{3.6}$$

$$|\psi_2\rangle = (X_1 \otimes X_3 2XOR_{i_1s_2o} X_1 \otimes X_3) |\psi_1\rangle \tag{3.7}$$

Quando aplicado ao circuito quântico, representado na Figura 3.5, os *Seletores* em superposição a partir da aplicação do operador *Hadamard* em cada um deles, teremos a

Equação 3.8, onde $i \in [0, 2]$.

$$|s_i\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^2 |i\rangle \quad (3.8)$$

A *qRAM2*, por sua vez, é formada pelos seguintes componentes: duas *Entradas*, quatro *Seletores* e uma *Saída*. Será iniciada pelos estados $ket\psi_{0_1,0_2,0_3,0_4,0_5,0_6,0_7}$ que serão manipulados pela *qRAM2* em ordem de funcionamento a partir dos seguintes passos regidos pelas Equações 3.9 3.10 3.11 3.12, as quais, com o decorrer do processo, realizam a partir de seu operadores a modificação do estado de *Saída*, representado por $|O_2\rangle$.:

$$|\psi_1\rangle = (3XOR_{i_1 i_2 s_3 o}) |\psi_0\rangle \quad (3.9)$$

$$|\psi_2\rangle = (I_1 \otimes X_2 3XOR_{i_1 i_2 s_2 o} X_2 \otimes I_1) |\psi_1\rangle \quad (3.10)$$

$$|\psi_3\rangle = (X_1 \otimes I_2 3XOR_{i_1 i_2 s_1 o} I_2 \otimes X_1) |\psi_2\rangle \quad (3.11)$$

$$|\psi_4\rangle = (X_1 \otimes X_2 3XOR_{i_1 i_2 s_0 o} X_2 \otimes X_1) |\psi_3\rangle \quad (3.12)$$

Os estados dos *Seletores* são iniciados por uma superposição proporcionada pelo operador *Hadamard* representado na Equação 3.8, em que $i \in [0, 4]$.:

$$|s_i\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^4 |i\rangle \quad (3.13)$$

Este trabalho realizou dois exemplos utilizando dois circuitos lógicos quânticos que representam uma Rede Neural Quântica Sem Peso por meio de um algoritmo.

O primeiro exemplo foi composto por uma múltipla Rede Neural Quântica Sem Peso, representada na Figura 4.1, que foi formada por meio do software Mathematica [22], que possui três qubits de entrada, variados pelo *Operador NOT* nos seguintes estados: $|000\rangle, |100\rangle, |110\rangle$.

Nos procedimentos iniciais foram aplicados operadores *Hadamard* nos *Seletores* para estes serem marcados nos respectivos acertos selecionados pelos *Operadores*. Cada *Operador* foi composto pela associação em paralelo de uma *qRAM1* e uma *qRAM2* com *Saídas* indicadas em série pela *qRAM2*, a qual aponta para um *Contador* correspondente.

Por fim, cada *Contador*, representado por C_n , irá corresponder a uma marcação após os respectivos acertos de saída com saída desejada, sendo a quantidade de contadores representada por $n \in [0, 2]$, a fim de que o operador formado pela *qRAM1* e *qRAM2*, mediante a aplicação dos *Operadores* de recuperação, recupere o estado inicial de *Saída*, de modo que esta seja representada por $|0\rangle$. Ressaltamos que esse processo deve ser realizado com cada operador.

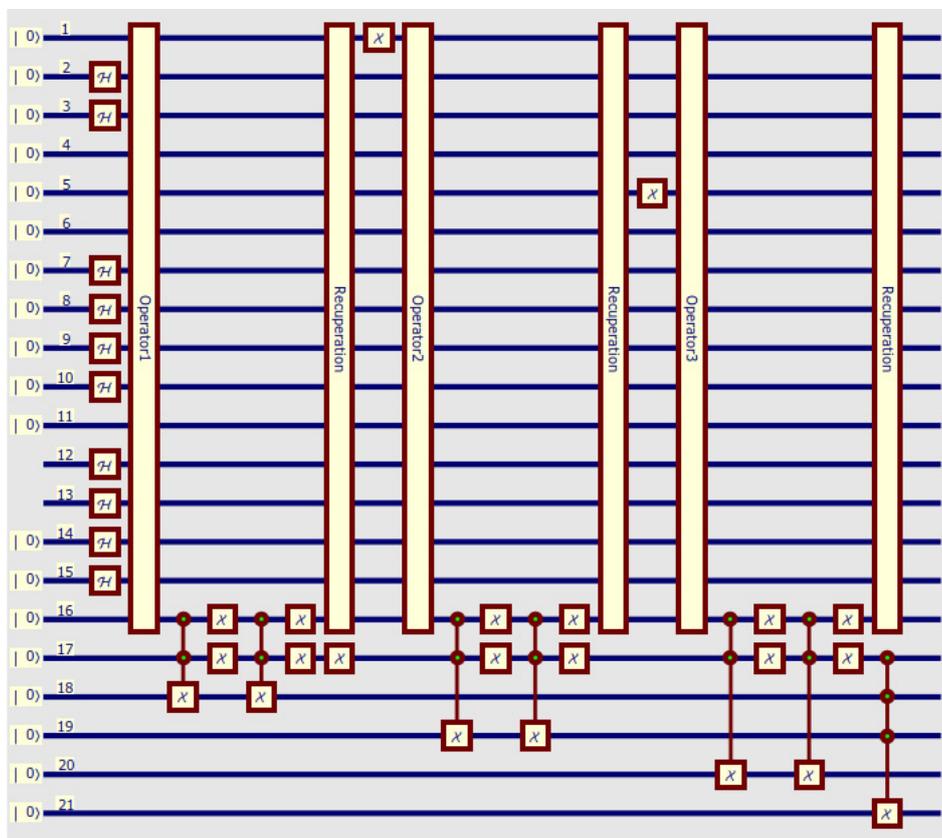


Figura 3.9: Figura de uma Rede Neural Quântica Sem Peso Completa [22] .

O segundo exemplo está sendo representado na Figura 4.6, que foi formada por meio do software Mathematica [22], o qual possui duas entradas padrões variadas ao longo do processo pelo o operador *NOT* com os seguintes valores $|00\rangle, |10\rangle, |11\rangle, |01\rangle$, sendo utilizado em cada processo uma *qRAM2* para a formação dos resultados de *Saída*.

Após cada processo é realizada uma marcação nos *Contadores* correspondentes as *Saídas* dos mesmos com a *Saída Desejada*, onde essa é composta pelo estado $d = |1011\rangle$, que está presente ao longo da linha 9 do circuito representado na Figura 4.6.

Ao final, cada *Contador*, representado por C_n , irá corresponder a uma marcação após os respectivos acertos de saída com saída desejada, sendo a quantidade de contadores representada por $n \in [0, 3]$, a fim de que o operador formado pela *qRAM2*, mediante a aplicação dos *Operadores* de recuperação, recupere o estado inicial de *Saída*, de modo que esta seja representada por $|0\rangle$. Ressaltamos que esse processo deve ser realizado com cada operador.

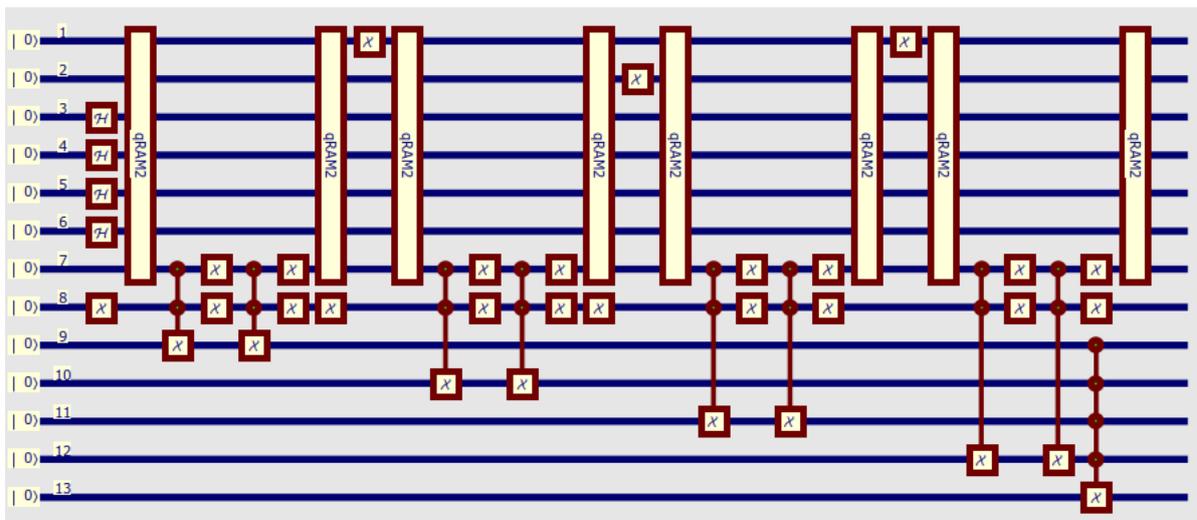


Figura 3.10: Figura de uma RNNQSP.

Capítulo 4

Algoritmo de Aprendizado Para o Treinamento de Redes Neurais Quânticas Sem Peso

Neste capítulo será apresentado, na seção 4.1 apresentação de uma transformação linear. Em seguida, seção 4.1.1 apresentação a descrição de um operador não-unitário. Posteriormente, em uma seção 4.2, o principal resultado desta dissertação, qual seja, o algoritmo de aprendizado que realiza transformação não-unitário linear para o treinamento de Redes Neurais Quânticas Sem Peso. As seções 4.3 e 4.4 apresentam exemplos da execução do circuito quântico utilizando valores numéricos de maneira detalhada.

4.1 Transformação Linear

As transformações lineares são manipulações algébricas que utilizam matrizes como ferramenta desse processo sendo tratado com mais detalhes no apêndice A.

4.1.1 Operadores Não-Unitários

Os operadores não-unitários A não possuem as propriedades dos operadores unitários U que seguem as seguintes afirmações $UU^{-1} = U^{-1}U = I$ e $U^\dagger = U^{-1}$ [2], onde I representa uma matriz identidade quadrática composta dos elementos da coluna principal com o valor igual a 1 e os demais 0.

Partindo de um conceito físico, a aplicação dos operadores unitários nos estados quânticos é dada pelo envio de informação e pela mudança destes estados sem ocorrer a perda de energia, tendo em vista que esta aplicação ocorre em um sistema isolado, sendo cada estado composto por um dado valor de energia.

Deste modo, o circuito quântico em seu pleno funcionamento teórico realiza um processo *cíclico* de energia [21], desde que o sistema seja fechado e livre de interferências externas.

Foi indicado na referência [11] a formação de circuitos quânticos para realizar transformações lineares utilizando, para isso, operadores não-unitários. Esses operadores não-unitários possivelmente podem resolver problemas da classe de complexidade computacional NP-COMPLETO [15] [32], o qual abrange os problemas de SAT3 [4], os problemas do Ciclo de Hamiltoniano [4] e facilitam as execuções dos resultados de Máquinas Registradoras Quânticas [15], entendidas como composições teóricas de circuitos lógicos quânticos que compõem a formação das Redes Neurais Quânticas Sem Peso com os seus contadores, conforme já descrito neste trabalho.

Os operadores não-unitários possuem a propriedade de ter como solução apenas os resultados satisfatórios do problema a ser encontrado, caso contrário, será proporcionada uma solução nula. Desta feita, os estados que não possuem soluções possíveis vão entrar em colapso e serão anulados pelo critério de ortogonalidade entre os estados ou vetores.

Isso pode ser demonstrado através da aplicação do operador $W = |01\rangle\langle 01| + |11\rangle\langle 11|$ a um estado $|\psi\rangle = |00\rangle + |01\rangle + |10\rangle + |11\rangle$, e pela operação algébrica a seguir, visando conservar os estados $|01\rangle$ e $|11\rangle$, e eliminar os demais estados.

$$W |\psi\rangle = (|01\rangle \langle 01| + |11\rangle \langle 11|) \cdot (|00\rangle + |01\rangle + |10\rangle + |11\rangle) = |01\rangle + |11\rangle \quad (4.1)$$

Este trabalho realizou o operador não-unitário 4.41 obtido por meio da seguinte referência [15] em um algoritmo, onde este operador não-unitário O proporciona a solução de problemas do tipo SAT3 [15], como também apresenta a solução de Máquinas Registradoras Quânticas [15], conforme demonstrado abaixo, n a quantidade de análises e o estado $|1\rangle$ será o componente principal do processo que selecionará todos os estados iguais a $|1\rangle$ no último qubit do circuito quântico. E nessa Equação foi realizada a adição do operador Identidade I , formando a Equação 4.3, por meio de uma manipulação de um produto tensorial com N representa o número de qubits de estrada do circuito. Este operador foi aplicado em dois exemplos proporcionando, assim, uma transformação linear no processo de execução.

$$O = 2^n |1\rangle \langle 1| \quad (4.2)$$

$$A = I^N \otimes 2^n |1\rangle \langle 1| \quad (4.3)$$

4.2 Algoritmo de aprendizado

Existem diversos tipos de algoritmos de treinamento para Redes Neurais Quânticas que utilizam operações não-lineares [25] [26]. Esse trabalho propõe a utilização de um algoritmo que utiliza operadores lineares para o treinamento de Redes Neurais Quânticas Sem Peso.

Nesse trabalho, utilizaremos um algoritmo, representado por Algoritmo 1, que supõe a viabilidade da aplicação de operações lineares quânticas não-unitárias. Assim, a maneira de aprendizado que está sendo apresentada pelo Algoritmo 1 utiliza um circuito quântico

composto de uma Rede Neural Quântica Sem Peso dada por q-PLN com uma $qRAM$ de tamanho qualquer e possui como estado de entrada $|\psi_i\rangle$, descrito pela Equação 4.4, em que $|i\rangle$ é o registro quântico de entrada, $|s\rangle$ são os *Seletores*, $|d\rangle$ a *Saída Desejada*, $|cont\rangle$ os contadores dos acertos e $|perf\rangle$ o estado de performance do circuito.

$$|\psi_i\rangle = (|i\rangle |s\rangle |o\rangle |d\rangle |cont\rangle) |perf\rangle \quad (4.4)$$

No algoritmo 1 a *linha 1* realiza o início do processo de entrada dos qubits, representados por $|i\rangle$, no registrador. Na *linha 2* são inicializados os qubits *Seletores*, representados por $|s\rangle$, aplicados a cada um deles um operador *Hadamard*. Desta maneira, os *Seletores*, representados por $|s\rangle$, terão seu formato apresentado pela Equação 4.5, onde o estado de *Saída* será representado por $|o\rangle$ na *linha 3*, a *Saída Desejada* representado por $|d\rangle$ na *linha 4* os *Contadores* por $|Cont\rangle$ na *linha 5* e o estado de *Performance* por $|0\rangle$ na *linha 6*.

$$|s\rangle = \left(\frac{1}{\sqrt{2}}\right)^{n_s} (|0\rangle + |1\rangle)^{\otimes n_s} \quad (4.5)$$

A *linha 7* é composta por um loop *for* que apresenta, para cada rodada, os padrões $|x\rangle$ que serão utilizados em todo o processo. Na *linha 8* cada padrão irá corresponder a um valor de i para a entrada e a *Saída Desejada*, representada por $|d\rangle$, será associada a um estado $|x\rangle$ na *linha 9*. Na *linha 10* será aplicada a $qRAM$ N genérica nos elementos das *linhas 8* e *9* para proporcionar uma *Saída*, representada por $|o\rangle$. Esse procedimento proporciona um estado dado pela Equação 4.6.

$$|\psi_n\rangle = \left(|i\rangle \left(\frac{1}{\sqrt{2}}\right)^{n_s} (|0\rangle + |1\rangle)^{\otimes n_s} |o\rangle |d(x)\rangle\right) |perf\rangle \quad (4.6)$$

Na *linha 11* será verificada se existe igualdade da *Saída* com relação ao padrão $|d(x)\rangle$. Sendo verificada essa igualdade, então adicione 1 ao estado $|cont\rangle$, o que corresponderá a cotação do laço *for*. Tal procedimento pode ser apresentado pela Equação 4.7.

Algorithm 1: Algoritmo de Aprendizado Não-Unitário Linear

```

1 Inicialize os qubits no registrador quântico  $|i\rangle$  com o estado  $|0\rangle$ 
2 Inicialize os qubits no Seletores  $|s\rangle$  com o estado quântico  $H|0\rangle$ 
3 Inicialize o qubit de saída  $|o\rangle$  com o estado quântico  $|0\rangle$ 
4 Inicialize o qubit de saída desejada  $|d\rangle$  com o estado quântico  $|0\rangle$ 
5 Inicialize os qubits no registrador  $|cont\rangle$  com o estado quântico  $|0\rangle$ 
6 Inicialize os qubits no registrador  $|perf\rangle$  com o estado quântico  $|0\rangle$ 
7 for Cada padrão  $x \in$  grupo de treinamento do
8    $|i\rangle \leftarrow |x\rangle$ 
9    $|d\rangle \leftarrow |d(x)\rangle$ 
10  Executar a  $qRAM$   $N$  para produzir o resultado no registro de saída  $|o\rangle$ 
11  if  $|Saída\rangle = |d(x)\rangle$  then
12    Adicionar 1 no registrador  $|cont_n\rangle$ 
13  end
14  Aplique o operador inverso a  $qRAM$   $N^{-1}$ 
15  Remova  $|x\rangle$  e  $|d(x)\rangle$  no registrador quântico  $|i\rangle$  e  $|d\rangle$ .
16 end
17 if  $|cont\rangle = |1\rangle_n$  then
18   Marque 1 no qubit performance  $|perf\rangle$ 
19 end
20 Aplicação do operador não-unitário  $A$  no registrador quântico  $|perf\rangle$ 

```

$$|\psi_n\rangle = \left(|i\rangle \left(\frac{1}{\sqrt{2}} \right)^{n_s} (|0\rangle + |1\rangle)^{\otimes n_s} |o\rangle |d(x)\rangle |1000\dots\rangle \right) |perf\rangle \quad (4.7)$$

Na *linha 14* será aplicado o operador inverso da $qRAM$ N^{-1} para restaurar o valor inicial da *Saída* que foi $|0\rangle$ apresentada pela *linha 15*. E o laço da *linha 7* será finalizado na *linha 16* até existir a quantidade para se realizar a troca dos estados de entrada $|x\rangle$ e dos padrões $|d(x)\rangle$ para que se chegue ao fim dos dados citados anteriormente.

A *linha 17* será composta por todos os estados dos *Contadores*, representados por $|cont\rangle$,

iguais ou não a 1, de modo que 1 será marcado no estado do qubit da *Performance*, representada por $|perf\rangle$ se todos os $|cont\rangle = 1$ caso contrário será 0 o estado do qubit da *Performance*, conforme mostrado a seguir:

$$|\psi_n\rangle = \left(|i\rangle \left(\frac{1}{\sqrt{2}} \right)^{n_s} (|0\rangle + |1\rangle)^{\otimes n_s} |o\rangle |d(x)\rangle |111\dots\rangle \right) |1\rangle. \quad (4.8)$$

Na *linha 20* será indicada a aplicação do operador não unitário A sobre o presente estado quântico dado na Equação 4.8 e em seguida serão realizados os passos da obtenção dos resultados.

$$A|\psi_n\rangle = A \left(\left(|i\rangle \left(\frac{1}{\sqrt{2}} \right)^{n_s} (|0\rangle + |1\rangle)^{\otimes n_s} |o\rangle |d(x)\rangle |1111\dots\rangle \right) |1\rangle \right) \quad (4.9)$$

$$A = I^N \otimes 2^n |1\rangle \langle 1| \quad (4.10)$$

No qual $N = 2^{m-1}$ e m é o número total de estados iniciais do sistema extraído a performance e $n = 0$ por só ter um estado para performance.

$$A|\psi_n\rangle = I^N \otimes 2^n |1\rangle \langle 1| \left(\sum_{i=1}^y a_n |\psi_1\rangle |1\rangle + \sum_{i=1}^z a_n |\psi_0\rangle |0\rangle \right) \quad (4.11)$$

$$A|\psi_n\rangle = I^N \cdot \left(\sum_{i=1}^y a_n |\psi_1\rangle \right) 2^n |1\rangle \langle 1| |1\rangle + I^N \cdot \left(\sum_{i=1}^z a_n |\psi_0\rangle \right) 2^n |1\rangle \langle 1| |0\rangle \quad (4.12)$$

No qual y é o número desconhecido de acertos e z o número desconhecido de erros.

$$A|\psi_n\rangle = I^N \cdot \left(\sum_{i=1}^y a_n |\psi_1\rangle \right) 2^n |1\rangle \langle 1| |1\rangle + 0 \quad (4.13)$$

$$A|\psi_n\rangle = I^N \cdot \left(\sum_{i=1}^y a_n |\psi_1\rangle \right) 2^n |1\rangle \langle 1| |1\rangle \quad (4.14)$$

Para uma melhor compreensão do Algoritmo 1 serão apresentados dois exemplos de seu funcionamento nas Seções 4.3 e 4.4, as quais representam simulações com problemas reais que não são possíveis de serem replicados em uma máquina convencional clássica devido ao custo exponencial para a simulação de um sistema quântico.

4.3 Exemplo 1

A rede neural quântica sem peso, formada pelo circuito quântico dado pela Figura 4.1, foi inicializada com 21 qubits todos no estado $|\varphi_0\rangle = |0\rangle$ e no decorrer do processo de cada análise os *qubits* de entrada terão os estados respectivos $|i\rangle = \{|000\rangle, |100\rangle \text{ e } |110\rangle\}$ variados por meio de operador *NOT*.

A construção da Figura 4.1 foi dada por meio do seguinte esquema do pacote *Quantum* apresentado na Figura 4.2.

4.3.1 Desenvolvimento algébrico do primeiro exemplo

Todo o processo de desenvolvimento foi realizado com os seguintes passos, dados pelo algoritmo 1:

1. Inicialize os qubits no registrador quântico $|i\rangle$ com o estado $|0\rangle$ na *linda 1*:

$$|\varphi_1\rangle = |0_1, 0_5, 0_6\rangle. \quad (4.15)$$

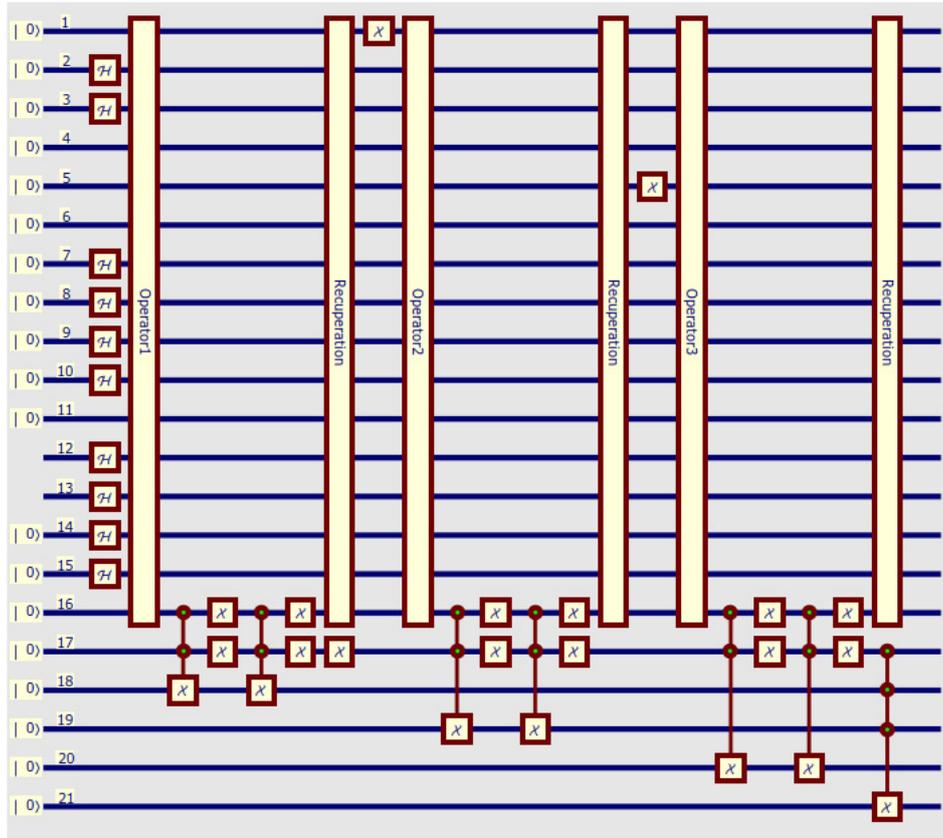


Figura 4.1: Figura de uma Rede Neural Quântica Sem Peso completa.

2. Inicialize os qubits no registrador $|s\rangle$ com o estado quântico $H|0\rangle$ na *linhas 2*:

$$|\varphi_1\rangle = (H_2 \otimes H_3 \otimes H_7 \otimes H_8 \otimes H_9 \otimes H_{10} \otimes H_{12} \otimes H_{13} \otimes H_{14} \otimes H_{15}) \cdot |0_2\rangle, |0_3\rangle, |0_7\rangle, |0_8\rangle, |0_9\rangle, |0_{10}\rangle, |0_{12}\rangle, |0_{13}\rangle, |0_{14}\rangle, |0_{15}\rangle. \quad (4.16)$$

3. Inicialize o qubit de saída $|o\rangle$ com o estado quântico $|0\rangle$ na *linha 3*:

$$|\varphi_1\rangle = |0_4, 0_{11}, 0_{17}\rangle. \quad (4.17)$$

```
QuantumPlot[
  c{i3, i4, i5}[X16] · (Recuperation1,2,3,4,5,6,7,8,9,10,11 · X11 ⊗ X12 · c{i1, i2}[X15] · X12 ⊗ X11 · c{i1, i2}[X15] · Operator1,2,3,4,5,6,7,8,9,10,11 · X5) ·
  (Recuperation1,2,3,4,5,6,7,8,9,10,11 · X11 ⊗ X12 · c{i1, i2}[X14] · X12 ⊗ X11 · c{i1, i2}[X14] · Operator1,2,3,4,5,6,7,8,9,10,11 · X1) ·
  (X12 ⊗ Recuperation1,2,3,4,5,6,7,8,9,10,11 · X11 ⊗ X12 · c{i1, i2}[X13] · X12 ⊗ X11 · c{i1, i2}[X13] · Operator1,2,3,4,5,6,7,8,9,10,11) ·
  H2 ⊗ H3 ⊗ H7 ⊗ H8 ⊗ H9 ⊗ H10 · |01, 02, 03, 04, 05, 06, 07, 08, 09, 010, 011, 012, 013, 014, 015, 016]
```

Figura 4.2: Figura da implementação da MRNQSP

4. Inicialize os qubits no registrador $|cont\rangle$ com o estado quântico $|0\rangle$ na *linha 5*:

$$|\varphi_1\rangle = |0_{18}, 0_{19}, 0_{20}\rangle. \quad (4.18)$$

5. Inicialize os qubits no registrador $|perf\rangle$ com o estado quântico $|0\rangle$ na *linha 6*:

$$|\varphi_1\rangle = |0_{21}\rangle. \quad (4.19)$$

6. Todos os estados quânticos que são manipulados pelo circuito quântico:

$$\begin{aligned} |\varphi_1\rangle = & |0_1\rangle \left(\frac{|0_2\rangle + |1_2\rangle}{\sqrt{2}} \right) \left(\frac{|0_3\rangle + |1_3\rangle}{\sqrt{2}} \right) |0_4\rangle |0_5\rangle |0_6\rangle \left(\frac{|0_7\rangle + |1_7\rangle}{\sqrt{2}} \right) \\ & \left(\frac{|0_8\rangle + |1_8\rangle}{\sqrt{2}} \right) \left(\frac{|0_9\rangle + |1_9\rangle}{\sqrt{2}} \right) \left(\frac{|0_{10}\rangle + |1_{10}\rangle}{\sqrt{2}} \right) |0_{11}\rangle \left(\frac{|0_{12}\rangle + |1_{12}\rangle}{\sqrt{2}} \right) \\ & \left(\frac{|0_{13}\rangle + |1_{13}\rangle}{\sqrt{2}} \right) \left(\frac{|0_{14}\rangle + |1_{14}\rangle}{\sqrt{2}} \right) \left(\frac{|0_{15}\rangle + |1_{15}\rangle}{\sqrt{2}} \right) |0_{16}\rangle |0_{17}\rangle |0_{18}\rangle \\ & |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \end{aligned} \quad (4.20)$$

7. Seleção dos resultados de entrada para o padrão correspondente nas *linha 7* e *linha 8* e aplicação do *Operador1* que é composta da $qRAM N$, apresentado pela Figura 4.3 *linha 10*, que é composto por uma $qRAM1$ e duas $qRAM2$. A seguir:

$$|\varphi_2\rangle = (qRAM_{1,2,3,4} \otimes qRAM_{2_{5,6,7,8,9,10,11}}) |\varphi_1\rangle \quad (4.21)$$

$$|\varphi_3\rangle = (qRAM_{2_{4,11,12,13,14,15,16}}) |\varphi_2\rangle \quad (4.22)$$

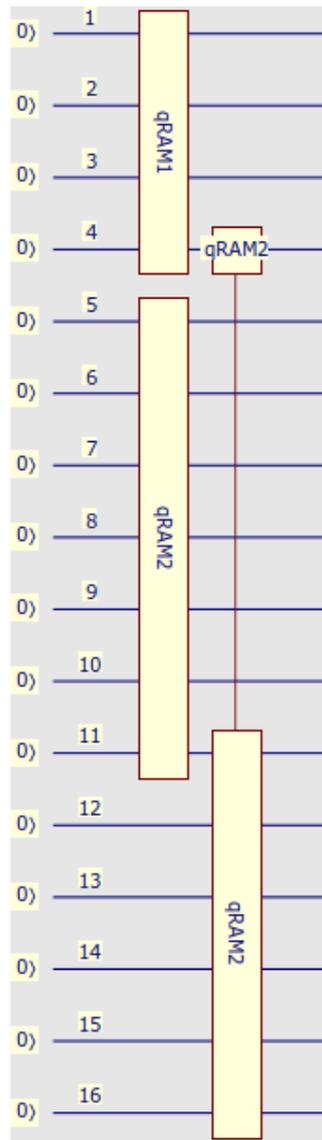


Figura 4.3: Operador1

8. Adicionar 1 no registrador $|cont_n\rangle p_{19}$ se $|0\rangle = |d\rangle$ nas *linha 11 e linha 12*:

$$|\varphi_4\rangle = (2XOR_{o_{16}d_{17}p_{18}})|\varphi_3\rangle \quad (4.23)$$

$$|\varphi_5\rangle = (X_{11} \otimes X_{17} \cdot 2XOR_{o_{11}d_{17}p_{18}} \cdot X_{17} \otimes X_{11})|\varphi_4\rangle \quad (4.24)$$

Apresentação do primeiro resultado obtido dado pela Equação 4.25:

$$\begin{aligned}
|\varphi_6\rangle = & \frac{1}{32}(|0_4\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |1_{13}\rangle |0_{14}\rangle |0_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |1_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |1_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |0_{16}\rangle |0_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |1_{13}\rangle |0_{14}\rangle |0_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |1_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |1_{11}\rangle |1_{21}\rangle |1_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |0_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle).
\end{aligned} \tag{4.25}$$

9. Aplicação do operador de recuperação da $qRAM$ N^{-1} nos estados anteriores *linha 14*:

$$|\varphi_6\rangle = (qRAM1_{\text{Recuperação}_{1,2,3,4}} \otimes qRAM2_{\text{Recuperação}_{5,6,7,8,9,10,11}}) |\varphi_5\rangle \tag{4.26}$$

$$|\varphi_7\rangle = (qRAM2_{\text{Recuperação}_{4,11,12,13,14,15,16}}) |\varphi_6\rangle \tag{4.27}$$

10. Seleção dos resultados de entrada para o padrão correspondente a *linha 7*, a *linha 8* e a aplicação do *Operador1* que é composta da *qRAM N*, apresentada pela Figura 4.3 *linha 10*, que é composta por uma *qRAM1* e duas *qRAM2*, como demonstrado a seguir:

$$|\varphi_8\rangle = [(qRAM1_{1,2,3,4} \otimes qRAM2_{5,6,7,8,9,10,11} \cdot X_{17} \cdot X_1)] |\varphi_7\rangle \quad (4.28)$$

$$|\varphi_9\rangle = (qRAM2_{4,11,12,13,14,15,16}) |\varphi_8\rangle \quad (4.29)$$

11. Adicionar 1 no registrador $|cont_n\rangle p_{19}$ se $|0\rangle = |d\rangle$ nas *linha 11* e *linha 12*:

$$|\varphi_{10}\rangle = (2XOR_{o_{11}} d_{17} p_{19}) |\varphi_9\rangle \quad (4.30)$$

$$|\varphi_{11}\rangle = (X_{11} \otimes X_{17} \cdot 2XOR_{o_{11}} d_{17} p_{19} \cdot X_{17} \otimes X_{11}) |\varphi_{10}\rangle \quad (4.31)$$

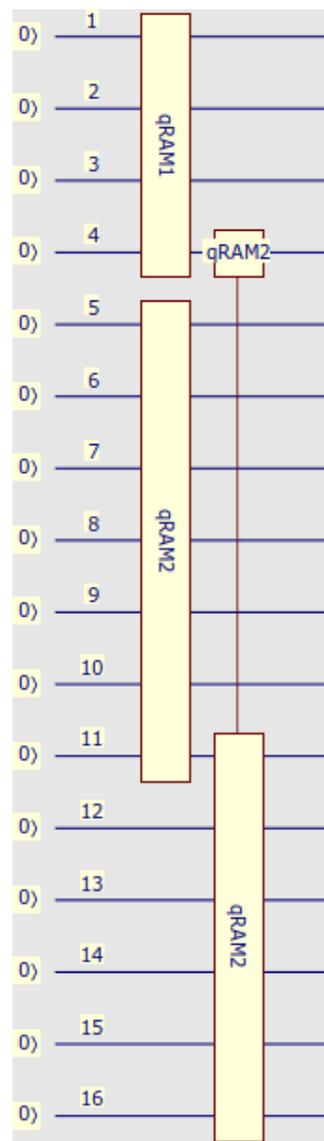


Figura 4.4: Operador1

Apresentação do primeiro resultado obtido dado pela Equação 4.32:

$$\begin{aligned}
 |\varphi_{11}\rangle = & \frac{1}{32}(|0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |1_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |1_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
 & + |0_4\rangle |0_{11}\rangle |1_{21}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle).
 \end{aligned} \tag{4.32}$$

12. Aplicação do operador de recuperação da $qRAM$ N^{-1} nos estados anteriores *linha 14*:

$$|\varphi_{12}\rangle = (qRAM1_{\text{Recuperação}_{1,2,3,4}} \otimes qRAM2_{\text{Recuperação}_{5,6,7,8,9,10,11}}) |\varphi_{11}\rangle \tag{4.33}$$

$$|\varphi_{13}\rangle = (qRAM2_{\text{Recuperação}_{4,11,12,13,14,15,16}}) |\varphi_{12}\rangle \tag{4.34}$$

13. Seleção dos resultados de entrada para o padrão correspondente a *linha 7*, a *linha 8* e a aplicação do *Operador1* que é composta da $qRAM$ N , apresentada pela Figura 4.3 *linha 10*, que é composta por uma $qRAM1$ e duas $qRAM2$, como a seguir:

$$|\varphi_{14}\rangle = (qRAM_{1,2,3,4} \otimes qRAM_{2_{5,6,7,8,9,10,11}} \cdot X_5) |\varphi_{13}\rangle \quad (4.35)$$

$$|\varphi_{15}\rangle = (qRAM_{2_{4,11,12,13,14,15,16}}) |\varphi_{14}\rangle \quad (4.36)$$

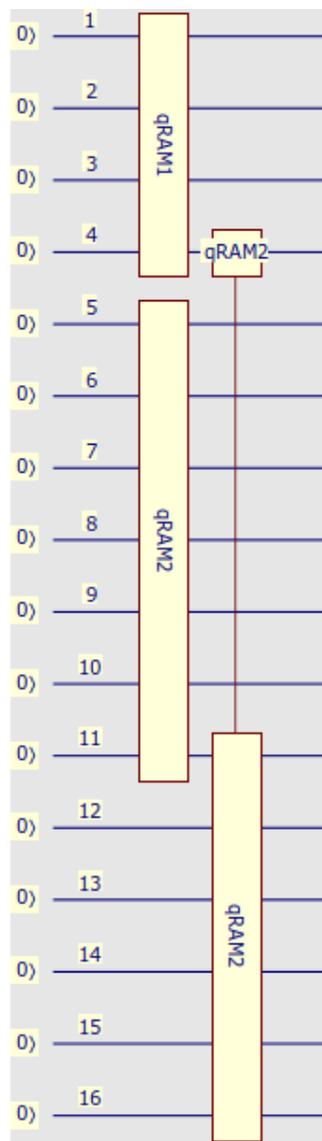


Figura 4.5: Operador1

14. Adicionar 1 no registrador $|cont_n\rangle p_{19}$ se $|0\rangle = |d\rangle$ nas *linha 11* e *linha 12*:

$$|\varphi_{16}\rangle = (2XOR_{o_{11}} d_{17} p_{20}) |\varphi_{15}\rangle \quad (4.37)$$

$$|\varphi_{17}\rangle = (X_{11} \otimes X_{17} \cdot 2XOR_{o_{11}} d_{17} p_{20} \cdot X_{17}) |\varphi_{16}\rangle \quad (4.38)$$

15. Marque 1 no qubit performance $|perf\rangle$, na *linha 18* e na *linha 17* aplicando um operador *Toffoli* nos *Contadores* estado $|cont\rangle$ para a marcação do estado de acerto do processo:

$$|\varphi_{18}\rangle = (Tofoli_{18,19,20,21}) |\varphi_{17}\rangle \quad (4.39)$$

Sendo apresentado a seguir o resultado de um determinado estado:

$$\begin{aligned}
|\varphi_{18}\rangle = & \frac{1}{32}(|0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |1_{20}\rangle |1_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |1_{20}\rangle |1_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |1_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |1_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |1_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |1_{20}\rangle |1_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |1_{20}\rangle |1_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |1_{13}\rangle |0_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |1_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |1_{20}\rangle |1_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{12}\rangle |1_{13}\rangle |1_{14}\rangle |0_{15}\rangle |0_{16}\rangle |1_{17}\rangle |0_{18}\rangle |1_{19}\rangle |0_{20}\rangle |0_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle |0_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |1_{18}\rangle |1_{19}\rangle |1_{20}\rangle |1_{21}\rangle \\
& + |0_4\rangle |0_{11}\rangle |1_{21}\rangle |1_{13}\rangle |1_{14}\rangle |1_{15}\rangle |1_{16}\rangle |1_{17}\rangle |0_{18}\rangle |0_{19}\rangle |0_{20}\rangle |0_{21}\rangle).
\end{aligned} \tag{4.40}$$

Após a extração dos resultados, sendo um deles representado pelo estado 4.40, este será aplicado a um operador não-unitário 4.41 o qual foi sugerido pela referência [15] que foi representada pela Equação 4.41 no algoritmo corresponde ao comando da *linha 20*:

$$A = I^N \otimes 2^n |1\rangle \langle 1|. \tag{4.41}$$

Os procedimentos resultaram de aplicações simples e diretas deste operador e o estado anteriormente citado foi dado por 4.42 4.43:

$$|\varphi_{Fim}\rangle = I^N \otimes 2^0 |1\rangle \langle 1| (|\varphi_{18}\rangle) \quad (4.42)$$

$$|\varphi_{Fim}\rangle = I^N \otimes 2^0 |1\rangle \langle 1| (|\varphi_{18_0}\rangle |1\rangle + |\varphi_{18_1}\rangle |0\rangle) \quad (4.43)$$

No qual $N = 2^{m-1}$ e m é o número total de estados iniciais do sistema extraído a performance da rede neural, e após esta aplicação serão apresentados os resultados satisfatórios, representados a seguir:

$$\begin{aligned} |\varphi_{Fim}\rangle = & \frac{1}{32} (|0_1\rangle |0_2\rangle |0_3\rangle |0_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle \\ & + |0_1\rangle |0_2\rangle |0_3\rangle |0_4\rangle |1_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle \\ & + |0_1\rangle |0_2\rangle |1_3\rangle |0_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle \\ & + |0_1\rangle |0_2\rangle |1_3\rangle |0_4\rangle |1_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle \\ & + |0_1\rangle |0_2\rangle |1_3\rangle |1_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |0_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle \\ & + |0_1\rangle |0_2\rangle |0_3\rangle |0_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle). \end{aligned} \quad (4.44)$$

4.4 Exemplo2

A rede neural quântica sem peso, formada pelo circuito quântico dado pela Figura 4.6, foi inicializada com 12 qubits todos no estado $|\varphi_0\rangle = |0\rangle$, e no decorrer do processo de cada análise os *qubits* de entrada terão os estados respectivos $|i\rangle = \{|00\rangle, |10\rangle, |11\rangle e |01\rangle\}$ variados por meio do operador *NOT*.

A construção da Figura 4.6 foi criada por meio do seguinte esquema do pacote *Quantum*, apresentado na Figura 4.7.

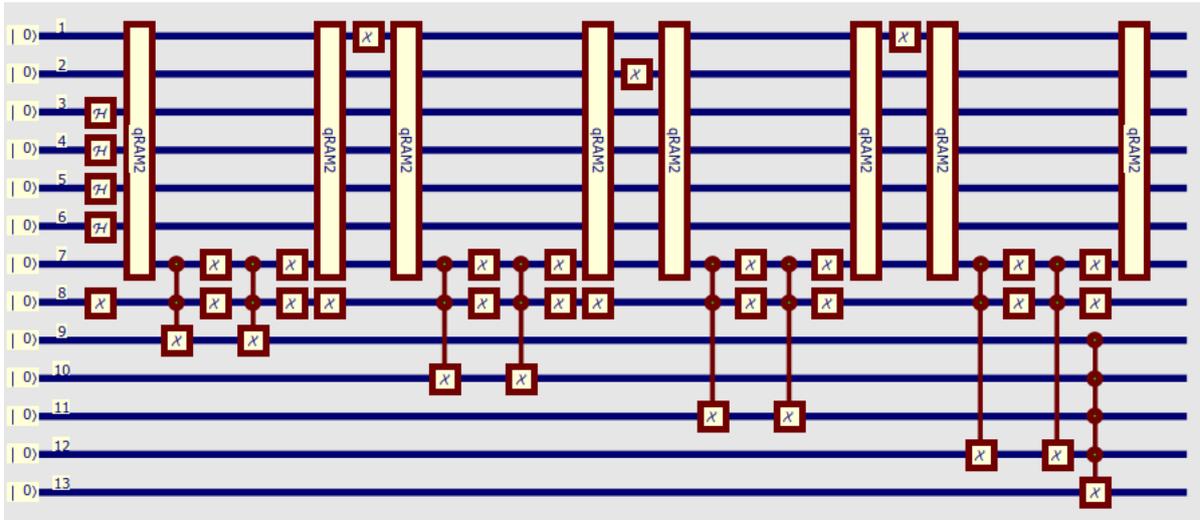


Figura 4.6: Figura de uma RNQSP.

$$\text{QuantumPlot} \left[\left(\text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_{12}] \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_{12}] \cdot \text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_1 \right) \cdot \right. \\ \left(\text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_{11}] \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_{11}] \cdot X_8 \cdot \text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_2 \right) \cdot \\ \left(\text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_{10}] \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_{10}] \cdot X_8 \cdot \text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_1 \right) \cdot \\ \left. \left(\text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_9] \cdot X_7 \otimes X_8 \cdot C^{\{\hat{8}, \hat{7}\}} [X_9] \cdot \text{qRAM2}_{\hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}, \hat{7}} \cdot X_8 \cdot H_3 \otimes H_4 \otimes H_5 \otimes H_6 \right) \cdot |0_1, 0_2, 0_3, \right. \\ \left. 0_4, 0_5, 0_6, 0_7, 0_8, 0_9, 0_{10}, 0_{11}, 0_{12} \right]$$

Figura 4.7: Figura da implementação da RNQSP

4.4.1 Desenvolvimento Algébrico do Segundo Exemplo

Todo o processo de desenvolvimento foi realizado com os seguintes passos relativamente ao algoritmo 1:

1. Inicialize os qubits no registrador quântico $|i\rangle$ com o estado $|0\rangle$ na *linha 1*:

$$|\psi_1\rangle = |0_1, 0_2\rangle. \quad (4.45)$$

2. Inicialize os qubits no registrador $|s\rangle$ com o estado quântico $H|0\rangle$ na *linhas 2*:

$$|\psi_1\rangle = (H_3 \otimes H_4 \otimes H_5 \otimes H_6) \cdot |0_3, 0_4, 0_5, 0_6\rangle. \quad (4.46)$$

Corresponde a um resultado:

$$\begin{aligned}
 |\psi_0\rangle = & \frac{1}{4}(|0000000100000\rangle) + \frac{1}{4}(|0000010100000\rangle) + \frac{1}{4}(|0000100100000\rangle) \\
 & + \frac{1}{4}(|0000110100000\rangle) + \frac{1}{4}(|0001000100000\rangle) + \frac{1}{4}(|0001010100000\rangle) \\
 & + \frac{1}{4}(|0001100100000\rangle) + \frac{1}{4}(|0001110100000\rangle) + \frac{1}{4}(|0010000100000\rangle) \\
 & + \frac{1}{4}(|0010010100000\rangle) + \frac{1}{4}(|0010100100000\rangle) + \frac{1}{4}(|0010110100000\rangle) \\
 & + \frac{1}{4}(|0011000100000\rangle) + \frac{1}{4}(|0011010100000\rangle) + \frac{1}{4}(|0011100100000\rangle) \\
 & + \frac{1}{4}(|0011110100000\rangle)
 \end{aligned} \tag{4.47}$$

3. Inicialize o qubit de saída $|o\rangle$ com o estado quântico $|0\rangle$ na *linha 3*:

$$|\psi_1\rangle = |0_7\rangle. \tag{4.48}$$

4. Inicialize os qubits no registrador $|cont\rangle$ com o estado quântico $|0\rangle$ na *linha 5*:

$$|\psi_1\rangle = |0_9, 0_{10}, 0_{11}, 0_{12}\rangle. \tag{4.49}$$

5. Todos os estados quânticos que são manipulados pelo circuito quântico:

$$\begin{aligned}
 |\psi_1\rangle = & |0_1\rangle |0_2\rangle \left(\frac{|0_3\rangle + |1_3\rangle}{\sqrt{2}} \right) \left(\frac{|0_4\rangle + |1_4\rangle}{\sqrt{2}} \right) \left(\frac{|0_5\rangle + |1_5\rangle}{\sqrt{2}} \right) \\
 & \left(\frac{|0_6\rangle + |1_6\rangle}{\sqrt{2}} \right) |0_7\rangle |0_8\rangle |0_9\rangle |0_{10}\rangle |0_{11}\rangle |0_{12}\rangle
 \end{aligned} \tag{4.50}$$

6. Inicialize os qubits no registrador $|perf\rangle$ com o estado quântico $|0\rangle$ na *linha 6*:

$$|\psi_1\rangle = |0_{13}\rangle. \tag{4.51}$$

7. Seleção dos resultados de entrada para o padrão correspondente nas *linha 7* e *linha 8* e aplicação do *Operador1* que é composta da *qRAM N*, apresentado pela Figura 4.3 *linha 10*, que é composto por uma qRAM2:

$$|\psi_2\rangle = (qRAM2_{1,2,3,4,5,7}) |\psi_1\rangle \quad (4.52)$$

8. Adicionar 1 no registrador $|cont_n\rangle p_9$ se $|0\rangle = |d\rangle$ nas *linha 11* e *linha 12*:

$$|\psi_3\rangle = (2XOR_{o_1d_1p_9}) |\psi_2\rangle \quad (4.53)$$

$$|\psi_4\rangle = (X_7 \otimes X_8 2XOR_{o_1d_1p_9} X_8 \otimes X_7) |\psi_3\rangle \quad (4.54)$$

9. Aplicação do operador de recuperação da *qRAM N*⁻¹ nos estados anteriores *linha 14*:

$$|\psi_5\rangle = (qRAM2_{Recuperação} \otimes X_8) |\psi_4\rangle \quad (4.55)$$

10. Seleção dos resultados de entrada para o padrão correspondente nas *linha 7* e *linha 8*, e aplicação do *Operador1* que é composto da *qRAM N*, apresentada pela Figura 4.3 *linha 10*, que é composta por uma qRAM2:

$$|\psi_6\rangle = (qRAM2_{1,2,3,4,5,6,7} \otimes X_1) |\psi_5\rangle \quad (4.56)$$

11. Adicionar 1 no registrador $|cont_n\rangle p_{10}$ se $|0\rangle = |d\rangle$ nas *linha 11* e *linha 12*:

$$|\psi_7\rangle = (2XOR_{o_7d_8p_{10}}) |\psi_6\rangle \quad (4.57)$$

$$|\psi_8\rangle = (X_7 \otimes X_8 2XOR_{o_7d_8p_{10}} X_8 \otimes X_7) |\psi_7\rangle \quad (4.58)$$

12. Aplicação do operador de recuperação da $qRAM N^{-1}$ nos estados anteriores *linha 14*:

$$|\psi_9\rangle = (qRAM2_{\text{Recuperação}} \otimes X_8) |\psi_8\rangle \quad (4.59)$$

13. Seleção dos resultados de entrada para o padrão correspondente nas *linha 7* e *linha 8*, e aplicação do *Operador1* que é composto da $qRAM N$, apresentada pela Figura 4.3 *linha 10*, composta por uma qRAM2:

$$|\psi_{10}\rangle = (qRAM2_{1,2,3,4,5,6,7} \otimes X_2) |\psi_9\rangle \quad (4.60)$$

14. Adicionar 1 no registrador $|cont_n\rangle p_{11}$ se $|0\rangle = |d\rangle$ nas *linha 11* e *linha 12*:

$$|\psi_{11}\rangle = (2XOR_{o_7d_8p_{11}}) |\psi_{10}\rangle \quad (4.61)$$

$$|\psi_{12}\rangle = (X_7 \otimes X_8 2XOR_{o_7d_8p_2} X_8 \otimes X_7) |\psi_{11}\rangle \quad (4.62)$$

15. Aplicação do operador de recuperação da $qRAM N^{-1}$ nos estados anteriores *linha 14*:

$$|\psi_{12}\rangle = (qRAM2_{\text{Recuperação}}) |\psi_{11}\rangle \quad (4.63)$$

16. Seleção dos resultados de entrada para o padrão correspondente nas *linha 7* e *linha 8*, e aplicação do *Operador1* que é composto da $qRAM N$, apresentado pela Figura 4.3 *linha 10*, que é composta por uma qRAM2:

$$|\psi_{13}\rangle = (qRAM2_{1,2,3,4,5,6,7} \otimes X_1) |\psi_{12}\rangle \quad (4.64)$$

17. Adicionar 1 no registrador $|cont_n\rangle p_{11}$ se $|0\rangle = |d\rangle$ nas *linha 11* e *linha 12*:

$$|\psi_{14}\rangle = (2XOR_{o_7d_8p_{12}})|\psi_{13}\rangle \quad (4.65)$$

$$|\psi_{15}\rangle = (X_7 \otimes X_8 2XOR_{o_7d_8p_{12}} X_8 \otimes X_7)|\psi_{14}\rangle \quad (4.66)$$

18. Aplicação do operador de recuperação da $qRAM$ N^{-1} nos estados anteriores *linha 14*:

$$|\psi_{16}\rangle = (qRAM2_{Recuperação})|\psi_{15}\rangle \quad (4.67)$$

19. Marque 1 no qubit performance $|perf\rangle$, na *linha 17*, aplicando um operador *Toffoli* nos *Contadores* de estado $|cont\rangle$ para a marcação do estado de acerto do processo:

$$|\psi_{17}\rangle = (Toffoli_{9,10,11,12,13})|\psi_{16}\rangle \quad (4.68)$$

$$\begin{aligned}
 |\psi_{17}\rangle = & \frac{1}{4}(|0_1\rangle |1_2\rangle |0_3\rangle |0_4\rangle |0_5\rangle |0_6\rangle |0_7\rangle |1_8\rangle |0_9\rangle |0_{10}\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |0_4\rangle |0_5\rangle |1_6\rangle |0_7\rangle |1_8\rangle |0_9\rangle |0_{10}\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |0_4\rangle |1_5\rangle |0_6\rangle |0_7\rangle |1_8\rangle |0_9\rangle |1_{10}\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |0_4\rangle |1_5\rangle |1_6\rangle |0_7\rangle |1_8\rangle |0_9\rangle |1_{10}\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |1_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |0_9\rangle |0_{10}\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |1_4\rangle |0_5\rangle |0_6\rangle |1_7\rangle |1_8\rangle |0_9\rangle |0_{10}\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |1_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |0_9\rangle |0_{10}\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |1_4\rangle |1_5\rangle |0_6\rangle |1_7\rangle |1_8\rangle |0_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |0_3\rangle |1_4\rangle |1_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |0_9\rangle |1_{10}\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_0\rangle |1_2\rangle |1_3\rangle |0_4\rangle |0_5\rangle |0_6\rangle |0_7\rangle |1_8\rangle |1_9\rangle |0_{10}\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |1_3\rangle |0_4\rangle |0_5\rangle |1_6\rangle |0_7\rangle |1_8\rangle |1_9\rangle |0_{10}\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |1_3\rangle |0_4\rangle |1_5\rangle |0_6\rangle |0_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |1_3\rangle |0_4\rangle |1_5\rangle |1_6\rangle |0_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |0_{11}\rangle |0_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |1_3\rangle |1_4\rangle |0_5\rangle |0_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |0_{10}\rangle |1_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_1\rangle |1_2\rangle |1_3\rangle |1_4\rangle |0_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |0_{10}\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle \\
 & + |0_0\rangle |1_2\rangle |1_3\rangle |1_4\rangle |1_5\rangle |0_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle |1_{13}\rangle \\
 & + |0_0\rangle |1_2\rangle |1_3\rangle |1_4\rangle |1_5\rangle |1_6\rangle |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |0_{11}\rangle |1_{12}\rangle |0_{13}\rangle)
 \end{aligned} \tag{4.69}$$

Com a posse do estado $|\psi_{17}\rangle$ foi realizada a aplicação do operador não-unitário 4.41. E este foi realizado da maneira seguinte:

$$|\psi_{Final}\rangle = A |\psi_{17}\rangle. \tag{4.70}$$

O qual apresentará o resultado seguinte:

$$|\psi_{Final}\rangle = \frac{1}{4} |1_7\rangle |1_8\rangle |1_9\rangle |1_{10}\rangle |1_{11}\rangle |1_{12}\rangle |1_{13}\rangle. \tag{4.71}$$

Onde todos os estados que não apresentaram solução satisfatória para a rede foram anulados.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste capítulo serão apresentadas as conclusões que foram obtidas pelo estudo e análise algébrica do problema, como também serão apresentados trabalhos futuros.

5.1 Conclusão

Em seções anteriores, foram apresentadas que nas literaturas [25] [26] existem diversos tipos de algoritmos de treinamento para Redes Neurais Quânticas que utilizam operações não-lineares. Por outro lado, este trabalho de dissertação foi possível apresentar um algoritmo que utiliza a ferramenta da rede neural quântica sem peso, utilizando a rede q-PLN, acoplada ao operador não-unitário.

A partir da associação desses dois elementos citados a cima proporcionou a realização de transformações lineares como o processo de treinamento da rede neural para o critério de seleção de resultados que possui características.

Este algoritmo possui um custo de aprendizagem na ordem de $O(n_s * n_t)$, proporciona a seleção de 100% dos acertos de todos os estados quânticos que se apresentem como possíveis soluções do problema. E algoritmo foi validado por meio de manipulações algébricas. Essa nova área de estudo pode proporcionar novas perspectivas, pois permite a associação do estudo das Redes Neurais Quânticas Sem Peso com as transformações lineares associadas aos

operadores não-lineares que proporcionam apenas aos estados convenientes ao procedimento.

5.2 Trabalhos Futuros

O algoritmo que esse trabalho de dissertação propõe pode proporcionar o auxílio no estudo das Memórias Quânticas Associativas [34]. Ele utiliza uma rede do tipo q -PLN associada a um operador não-unitário acoplado a rede formado com o auxílio da Equação 4.41, que seleciona apenas os resultados que serão possíveis soluções em superposição a q -PLN. Unindo essas duas ferramentas possibilitam uma verificação rápida e direta de quais estados serão armazenados pelas Memórias Quânticas Associativas dado pelo seu contador. No qual este indicará os reais estados que serão armazenados na memória e recuperados.

Para uma outra maneira de aplicar o algoritmo proposto seria na realização de buscas de resultados sendo assim associado ao algoritmo de Grover [12]. Como o algoritmo de Grover realiza a seleção dos estados por meio da manipulação de suas amplitudes o algoritmo proposto aqui irá receber todos os resultados do algoritmo de Grover e anular o estado que não for solução desejada do processo.

Referências Bibliográficas

- [1] I. Aleksander. Self-adaptive universal logic circuits. *Electronics Letters*, 1966.
- [2] George B Arfken and Hans J Weber. *Mathematical methods for physicists*. Elsevier, 1922.
- [3] P. Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563–591.
- [4] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.
- [5] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir. Classical and superposed learning for quantum weightless neural networks. *Neurocomputing*, 75:52–60, 2012.
- [6] R. L. de Carvalho, P. M. V. Lima D. Carvalho, F. Mora-Camino, and F. M. G. Francsa. Online tracking of multiple objects using wisard. *European Symposium on Artificial Neural Networks*, page 541–546, 2014.
- [7] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [8] J. Faber and G. A. Giraldi. Quantum models for artificial neural networks. *Electronically available: <http://arquivosweb.incc.br/pdfs/QNN-Review.pdf>*, 2002.
- [9] Richard P. Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.

- [10] E. Fredkin and T. Toffoli. Conservative logic. *Int. J. Theor. Phys.*, 21:219–253, 1982.
- [11] Robert M. Gingrich and Colin P. Williams. Non-unitary probabilistic quantum computing. In *Proceedings of the Winter International Symposium on Information and Communication Technologies, WISICT '04*, pages 1–6. Trinity College Dublin, 2004.
- [12] Lov K Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical review letters*, 79(2):325, 1997.
- [13] S. Haykin. *Redes Neurais: Princípios e prática*. Bookman, Porto Alegre-RJ, 2008.
- [14] Alberto Leporati and Sara Felloni. Three "quantum" algorithms to solve 3-sat. *Theor. Comput. Sci.*, 372(2-3):218–241, 2007.
- [15] Alberto Leporati and Sara Felloni. Three "quantum" algorithms to solve 3-sat. *Theoretical Computer Science*, 372(2 - 3):218 – 241, 2007.
- [16] Cheng-Kai Lu, Yi-Che Huang, and Cheng-Jung Lee. Adaptive guidance system design for the assistive robotic walker. *Neurocomputing*, 170:152 – 160, 2015. Advances on Biological Rhythmic Pattern Generation: Experiments, Algorithms and Applications- Selected Papers from the 2013 International Conference on Intelligence Science and Big Data Engineering (IScIDE 2013) Computational Energy Management in Smart Grids.
- [17] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.
- [18] Risto Miikkulainen. Evolving neural networks. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '10*, pages 2441–2460, New York, NY, USA, 2010. ACM.
- [19] M. G. Mini. Neural network based classification of digitized mammograms. In *Proceedings of the Second Kuwait Conference on e-Services and e-Systems, KCESS '11*, pages 2:1–2:5, New York, NY, USA, 2011. ACM.
- [20] Vicente Moret-Bonillo. Can artificial intelligence benefit from quantum computing? *Progress in Artificial Intelligence*, 3(2):89–105, 2015.

-
- [21] Michael A. Nielsen, Isaac L. Chuang, and Ivan S Oliveira. *Computação quântica e informação quântica*. Bookman, 2005.
- [22] José Luis Gómez-Muñoz and Francisco Delgado. Quantum: A free mathematica add-on for dirac bra-ket notation, quantum algebra, quantum computing and the qhd approximation to the heisenberg equations of motion, November 2012.
- [23] W. Oliveira, A. J. Silva, T. B. Ludermir, A. Leonel, W. Galindo, and J. Pereira. Quantum logical neural networks. *Neural Networks*, pages 147–152, 2008.
- [24] W. Oliveira, A.J. Silva, T.B. Ludermir, and A. Leonel. Quantum logical neural networks. In *Neural Networks, 2008. SBRN '08. 10th Brazilian Symposium on*. IEEE.
- [25] M. Panella and G. Martinelli. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications*, 39:61–77, 2009.
- [26] Massimo Panella and Giuseppe Martinelli. Neurofuzzy networks with nonlinear quantum learning. *Fuzzy Systems, IEEE Transactions on*, 17(3):698–710, 2009.
- [27] Renato Portugal, Carlile C Lavor, Luiz M Carvalho, and Nelson Maculan. *Uma introdução à computação quântica*. Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), 2012.
- [28] P Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Peer Reviewed Journal*, 65(6):386–408, 1958.
- [29] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13(11):2567–2586, November 2014.
- [30] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.
- [31] Adenilton J Silva, Wilson R Oliveira, and Tereza L Ludermir. Classical and superposed learning for quantum weightless neural networks. *Neurocomputing*, 75:52–60, 2012.
- [32] Michael Sipser. *Introduction to the theory of computation*. Thomson Course Technology, Boston, 2006.

-
- [33] T. Toffoli. Reversible computing. *MIT Technical Report MIT/LCS/TM-151*, 1980.
- [34] C. Trugenberger. Probabilistic quantum memories. *Phys. Rev. Lett.*, 87(6):p. 067901, Jul 2001.

Apêndice A

Álgebra Linear

A.1 Vetores

A álgebra linear é a ferramenta matemática utilizada na Mecânica Quântica moderna. Comumente a representação é através da notação de Dirac (braket $|\cdot\rangle$) com o intuito de direcionar a aplicação matemática para o conhecimento físico. E através desta álgebra, utilizamos vetores para apresentar e manipular estados quânticos. Um vetor $\vec{a} = (a_1, a_2, \dots, a_n)$ é representado na notação de Dirac em formato ket:

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \tag{A.1}$$

sendo n um número inteiro positivo. E para a manipulação da soma e produto destes vetores por escalares reais \mathfrak{R} temos respectivamente a descrição $|a\rangle + |b\rangle$ e $k|a\rangle$ são apresentadas nas Equações A.2A.3:

Soma de vetores

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, |b\rangle = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \Rightarrow |a\rangle + |b\rangle = \begin{pmatrix} a_1 + b_1 \\ a_1 + b_2 \\ \vdots \\ a_n + b_n \end{pmatrix} \quad (\text{A.2})$$

Produto de vetor por escalar k

$$k(|a\rangle) = k|a\rangle = \begin{pmatrix} ka_1 \\ a_2 \\ \vdots \\ ka_n \end{pmatrix} \quad (\text{A.3})$$

A.2 Bra $\langle\psi|$

Na existência dos vetores de estados *kets* há um vetor de estado dual associado a ele chamado de *bra* $\langle\psi|$. De modo que, o vetor *bra* $\langle\psi|$ é um correspondente dual $|\psi\rangle$:

$$(|\psi\rangle^*)^T = \langle\psi|. \quad (\text{A.4})$$

A.3 Combinação Linear

Seja \mathbf{V} um espaço vetorial complexo finito contendo os vetores $|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle$. Seja a_1, a_2, \dots, a_n números complexos, então uma combinação linear em \mathbf{V} pode ser definido por:

$$a_1 |v_1\rangle + a_2 |v_2\rangle + \dots + a_n |v_n\rangle = \sum_{i=1}^n a_i |v_i\rangle \quad (\text{A.5})$$

Uma combinação linear pode ser classificada em Linearmente Dependente (LD) ou Linearmente Independente (LI). Para classificar em LI a combinação linear possui os valores a_1, a_2, \dots, a_n dada por meio da expressão a seguir:

$$a_1 |v_1\rangle + a_2 |v_2\rangle + \dots + a_n |v_n\rangle = 0 \quad (\text{A.6})$$

Para uma combinação Linear ser LD, então tem que existir a desigualdade $a_n \neq 0$

A.4 Produto Interno

É uma manipulação matemática que é utilizada por dois vetores $|v\rangle$ e $|u\rangle$ que proporciona um número. Deste modo, o produto interno é constituído pelos seguintes tópicos:

- Se $\langle v|u\rangle = 0$, então os vetores $|v\rangle$ e $|u\rangle$ são ortogonais.
- Para $\langle v|u\rangle \geq 0$, com $|v\rangle \geq 0$.
- $\langle v|u\rangle^\dagger = \langle u|v\rangle$.

Pode-se obter a norma de um vetor $\| |v\rangle \|$ através de $\sqrt{\langle v|v\rangle}$. Para *normalizar* $|v\rangle$ é utilizado Equação:

$$|\tilde{V}\rangle = \left(\frac{1}{\sqrt{\langle v|u\rangle}} \right) \cdot |v\rangle \quad (\text{A.7})$$

se $\| |v\rangle \| = 1$, este é dito ser unitário.

A.5 Operadores Lineares

Os operadores lineares são utilizados para realizar uma possível transformação de vetores. Dado um vetor $|v\rangle$, sendo aplicado a um operador A , gera-se a um novo vetor com o formato $|w\rangle = A|v\rangle$. Então, o operador linear entre os espaços vetoriais \mathbf{V} e \mathbf{W} proporciona uma transformação linear representada por $A : V \rightarrow W$. E a apresentação dos operadores lineares são também representados por meio de matrizes do tipo quadradas com formação $2^n \times 2^n$, onde n é a dimensão do sistema.

A.6 Produto Externo

O produto externo é realizado por dois vetores de estados, um ket $|\psi\rangle$ e o outro bra $\langle\varphi|$, que proporciona a formação de um operador como mostrado na Equação A.10:

$$A = |\psi\rangle \langle\varphi|. \quad (\text{A.8})$$

A demonstração matricial algébrica da formação de um produto externo é dado por meio de:

$$|\psi\rangle = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix}, \quad \langle\varphi| = \begin{pmatrix} b_{11} & b_{12} \end{pmatrix} \quad (\text{A.9})$$

$$A = |\psi\rangle \langle\varphi| = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} \\ a_{21}b_{11} & a_{21}b_{12} \end{pmatrix} \quad (\text{A.10})$$

A.7 Produto Tensorial

O Produto tensorial é uma forma de se unir espaços vetoriais ou vetores entre si para formar espaços vetoriais maiores ou vetores maiores, respectivamente. [21]

Propriedades:

- Se as dimensões dos espaços vetoriais \mathbf{V} e \mathbf{W} são respectivamente $\dim(V) = N_1$ e $\dim(W) = N_2$, a dimensão de um produto tensorial é $\dim(V \otimes W) = N_1 N_2$.
- Multiplicação de um escalar k em um produto tensorial entre estados $|\psi\rangle$ e $|\varphi\rangle$:

$$k(|\psi\rangle \otimes |\varphi\rangle) = (k|\psi\rangle) \otimes |\varphi\rangle = |\psi\rangle \otimes (k|\varphi\rangle) \quad (\text{A.11})$$

- Se $|\psi\rangle \in V$ e $|\varphi_1\rangle, |\varphi_2\rangle \in W$,

$$|\psi\rangle \otimes (|\varphi_1\rangle + |\varphi_2\rangle) = |\psi\rangle \otimes |\varphi_1\rangle + |\psi\rangle \otimes |\varphi_2\rangle. \quad (\text{A.12})$$

- Se $(|\psi_1\rangle, |\psi_2\rangle) \in V$ e $|\varphi\rangle \in W$

$$(|\psi_1\rangle + |\psi_2\rangle) \otimes |\varphi\rangle = |\psi_1\rangle \otimes |\varphi\rangle + |\psi_2\rangle \otimes |\varphi\rangle. \quad (\text{A.13})$$

É simples a representação de um produto tensorial por meio de matrizes pois é utilizado o esquema intitulado de *Produto de Kronecker* [21]. Para duas matrizes um A com dimensões $n \times m$ e uma outra matriz B com dimensões $p \times q$, são manipuladas da seguinte maneira:

$$A_{m \times n} = \begin{pmatrix} a_{1,1} & a_{1,\dots} & a_{1,n} \\ a_{\vdots,1} & \ddots & a_{2,n} \\ a_{m,1} & a_{m,\dots} & a_{m,n} \end{pmatrix}, B_{p \times q} = \begin{pmatrix} b_{1,1} & b_{1,\dots} & b_{1,q} \\ b_{\vdots,1} & \ddots & b_{2,q} \\ b_{p,1} & b_{p,\dots} & b_{p,q} \end{pmatrix} \quad (\text{A.14})$$

$$A_{m \times n} \otimes B_{p \times q} = \begin{pmatrix} a_{1,1}B_{p \times q} & a_{1,\dots}B_{p \times q} & a_{1,n}B_{p \times q} \\ a_{\cdot,1}B_{p \times q} & \ddots & a_{2,n}B_{p \times q} \\ a_{m,1}B_{p \times q} & a_{m,\dots}B_{p \times q} & a_{m,n}B_{p \times q} \end{pmatrix} \quad (\text{A.15})$$

Dado o *Produto Tensorial* entre os estados $|\psi\rangle$ e $|\varphi\rangle$ representados a seguir:

$$|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix}, |\varphi\rangle = \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} \quad (\text{A.16})$$

É representado o procedimento do *Produto Tensorial* por vetores colunas $|\psi\rangle$ e $|\varphi\rangle$:

$$|\gamma\rangle = |\psi\rangle \otimes |\varphi\rangle \quad (\text{A.17})$$

$$|\gamma\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} \otimes \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} = \begin{pmatrix} \psi_1 \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} \\ \psi_2 \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \psi_1\varphi_1 \\ \psi_1\varphi_2 \\ \psi_2\varphi_1 \\ \psi_2\varphi_2 \end{pmatrix} \quad (\text{A.18})$$