



Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

Redes Neurais Lógicas Quânticas

Adenilton José da Silva

Dissertação de Mestrado

Recife
21 de fevereiro de 2011

Universidade Federal de Pernambuco
Centro de Informática

Adenilton José da Silva

Redes Neurais Lógicas Quânticas

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Uni-
versidade Federal de Pernambuco como requisito parcial
para obtenção do grau de Mestre em Ciência da Com-
putação.*

Orientadora: *Teresa Bernarda Ludermir*
Co-orientador: *Wilson Rosa de Oliveira*

Recife
21 de fevereiro de 2011

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Silva, Adenilton José da
Redes neurais lógicas quânticas / Adenilton José da Silva -
Recife: O Autor, 2011.
xi, 88 folhas: il., fig.

Orientador: Teresa Bernarda Ludermir.
Dissertação (mestrado) Universidade Federal de
Pernambuco. Cln. Ciência da Computação, 2011.

Inclui bibliografia.

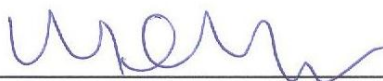
1. Redes neurais artificiais. 2. Computação quântica. 3.
Redes neurais quânticas. I. Ludermir, Teresa Bernarda
(orientadora). II. Título.

006.3

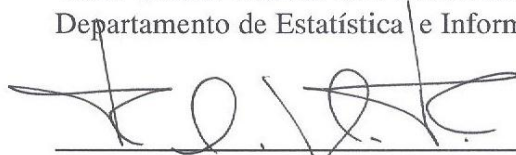
CDD (22. ed.)

MEI2011 – 025

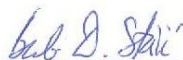
Dissertação de Mestrado apresentada por **Adenilton José da Silva** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Redes Neurais Lógicas Quânticas**”, orientada pela **Profa. Teresa Bernarda Ludermir** e co-orientada pelo **Prof. Wilson Rosa de Oliveira Junior** e aprovada pela Banca Examinadora formada pelos professores:



Prof. Wilson Rosa de Oliveira Junior
Departamento de Estatística e Informática / UFRPE



Prof. Felipe Maia Galvão França
Programa de Engenharia de Sistemas / UFRJ



Prof. Borko Stosic
Departamento de Estatística e Informática / UFRPE

Visto e permitida a impressão.
Recife, 21 de fevereiro de 2011.



Prof. Nelson Souto Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

A minha família.

Agradecimentos

Agradeço a todos que contribuíram no desenvolvimento desse trabalho. Especialmente a professora e orientadora Dra. Teresa Bernarda Ludermir e ao professor e co-orientador Dr. Wilson Rosa de Oliveira Junior pelo apoio e incentivo acadêmico e pessoal.

A todos os colegas de turma e de laboratório pelas incontáveis horas de estudo em conjunto e pelos momentos de descontração.

À minha família pelo apoio e compreensão.

Ao CNPq e FACEPE pelo suporte financeiro.

*Le monde de la réalité a ses limites; le monde de l'imagination est sans
frontières.*

—JEAN JACQUES ROUSSEAU

Resumo

Através da miniaturização dos componentes dos chips a cada ano a velocidade dos computadores é aproximadamente duplicada. Esta rápida redução dos componentes dos chips é conhecida como a Lei de Moore. Apesar de se manter verdadeira nos últimos anos, a lei de Moore está se aproximando de seu limite, pois os componentes dos chips estão se aproximando a escala atômica. Neste momento, será necessário considerar os efeitos da mecânica quântica sobre a computação.

O estudo dos modelos de computação não convencionais, como a computação quântica, é um dos grandes desafios da pesquisa em computação no Brasil. O desenvolvimento de novos hardwares com tecnologias diferentes do silício pode ter consequências nas técnicas de desenvolvimento de hardware e software.

O objetivo desta dissertação é investigar que vantagens podem ser obtidas através da aplicação de técnicas da computação quântica no desenvolvimento e treinamento de modelos de redes neurais artificiais.

Três modelos de redes neurais quânticas baseados em modelos de redes neurais sem pesos foram propostos. Ao contrário dos outros modelos de redes neurais quânticas, as redes propostas nesta dissertação podem simular as redes em que foram baseadas.

A principal vantagem dos modelos quânticos neurais propostos nesta dissertação está no seu algoritmo de treinamento, um algoritmo onde a rede neural é executada apenas uma vez independente do tamanho do conjunto de treinamento e da rede neural. O algoritmo proposto foi baseado em uma memória associativa quântica e no algoritmo de busca de Grover.

Palavras-chave: Redes Neurais sem Pesos, Computação Quântica, Redes Neurais Quânticas

Abstract

The computers have approximately doubled its velocity in the last years and this impressive progression in technology occurs due to of miniaturization of chips, knew as Moore's Law. The Moore's Law is near of its limit because the components of the chips are near of atomic scale. In this moment will be necessary consider the quantum mechanical effects on computing.

The study of the nonconventional models of computation, as the quantum computation, is one of Grand Challenges in Computer Science Research in Brazil. The development of new hardware with technologies different of silicon can have consequences in the development techniques of hardware and software.

The objective of this essay is to investigate what advantages one can to obtain using quantum computing techniques in the development and training of artificial neural networks models.

Three models of quantum neural networks based on weightless neural networks were proposed. Unlike other models of quantum neural networks, the networks proposed in this dissertation can simulate networks that were based.

The main advantage of quantum weightless neural networks is its learning algorithm. A learning algorithm where one needs to run the network only twice is proposed. The proposed algorithm is based on a quantum associative memory and Grover's search algorithm.

Keywords: Quantum Computation, Neural Network, Quantum Neural Networks

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	4
1.3	Organização da dissertação	5
2	Álgebra Linear	6
2.1	Conjuntos Geradores, Independência Linear e Bases	7
2.2	Operadores Lineares	8
2.3	Representação de Transformações Lineares por Matrizes	10
2.4	Produto Interno	11
2.5	Autovalores e Autovetores	12
2.6	Adjunta e Operadores Hermitianos	12
2.7	Produto Tensorial	13
3	Computação Quântica	15
3.1	Introdução	15
3.2	Postulados da Mecânica Quântica	16
3.3	Qubits	17
3.4	Operações sobre qubits	19
3.5	Medição	21
3.6	Múltiplos Qubits	22
3.7	Paralelismo	23
3.8	Não clonagem	24
3.9	Circuitos Quânticos	25

3.9.1	Teleporte quântico	26
3.10	Algoritmos Quânticos	30
3.10.1	Algoritmo de Deutsch	30
3.10.2	Algoritmo de Busca	32
4	Memória quântica probabilística	36
4.1	Introdução	36
4.2	Armazenamento	37
4.3	Recuperação	40
4.4	Sumário do capítulo	44
5	Redes Neurais Artificiais	45
5.1	Introdução	45
5.2	Nodo RAM	46
5.3	Rede RAM	47
5.4	Nodo Lógico Probabilístico (PLN)	48
5.5	Rede PLN	50
5.6	Nodo Lógico Probabilístico Multivalorado (MPLN)	51
5.7	Nodo GSN	53
5.7.1	Estado de validação	53
5.7.2	Estado de Aprendizagem	54
5.7.3	Estado de uso	54
5.8	Rede GSN	54
5.9	Pré-processamento	55
5.10	Sumário do capítulo	56
6	Redes Neurais Quânticas	57
6.1	Introdução	57
6.2	Algoritmos iterativos	58
6.2.1	Neurônio artificial quântico	58
6.2.2	Regra delta quântica	59

6.2.3	Rede neural quântica M-P	61
6.3	Algoritmos baseados na superposição	63
6.4	Sumário do capítulo	64
7	Neurônios Lógicos Quânticos	65
7.1	qPLN	65
7.2	qMPLN	68
7.3	qRAM	70
7.3.1	Simulação dos modelos de redes neurais sem peso clássicos	72
7.3.2	Algoritmo de Aprendizado Baseado na Superposição (AAS)	73
7.3.3	Complexidade do algoritmo AAS	75
7.3.4	Treinamento do neurônio qRAM com o AAS	76
7.4	Implementando um conjunto universal de operadores	78
7.5	Sumário do capítulo	79
8	Conclusão	81
8.1	Considerações finais	81
8.2	Contribuições deste trabalho	81
8.3	Trabalhos Futuros	82

Lista de Figuras

3.1	Representação de um qubit na Esfera de Bloch	19
3.2	Circuito operador Hadamard	25
3.3	Símbolo circuito de medição	25
3.4	Circuito operador C_{NOT}	26
3.5	Outro representação circuito operador C_{NOT}	26
3.6	Circuito para construção de pares EPR	27
3.7	Circuito quântico para teleportar um qubit	29
3.8	Circuito quântico do algoritmo de Deutsch	32
4.1	Circuito armazenamento memória probabilística quântica	39
4.2	Circuito recuperação memória probabilística quântica	42
5.1	Nodo RAM	47
5.2	Discriminador	49
5.3	Rede PLN piramidal	50
7.1	Esquema neurônio qPLN	66
7.2	Neurônio qPLN em circuito quântico	67
7.3	Neurônio qPLN com seletores	68
7.4	qRAM Node	72
7.5	NLQ simulando Hadamard	79
7.6	NLQ simulando operador Toffoli	79

CAPÍTULO 1

Introdução

Neste capítulo é apresentada a motivação para o trabalho com redes neurais quânticas sem peso. A seção 1.1 apresenta um breve histórico sobre redes neurais e computação quântica, apontando alguns dos problemas dos modelos de redes neurais quânticas com peso. Na seção 1.2 são descritos os objetivos desta dissertação. Na seção 1.3 são descritos os capítulos desta dissertação.

1.1 Motivação

Através da miniaturização dos componentes dos chips a cada ano a velocidade dos computadores é aproximadamente duplicada. Esta rápida redução dos componentes dos chips é conhecida como a Lei de Moore. Apesar de se manter verdadeira nos últimos anos, a lei de Moore está se aproximando de seu limite, pois os componentes dos chips estão se aproximando a escala atômica. Neste momento, será necessário considerar os efeitos da mecânica quântica sobre a computação [34].

O estudo dos modelos de computação não convencionais, como a computação quântica, é um dos grandes desafios da pesquisa em computação no Brasil [1]. O desenvolvimento de novos hardwares com tecnologias diferentes do silício pode ter consequências nas técnicas de desenvolvimento de hardware e software.

As redes neurais artificiais têm sido estudadas há mais de 50 anos e atualmente existem diversos modelos aplicados com sucesso a uma grande variedade de problemas. Os primeiros modelos de neurônios artificiais, ou a *primeira geração*, possuíam como característica comum uma saída binária como, por exemplo, o nodo MCP proposto por McCulloch e Pitts [31] ou as redes neurais sem peso [4]. Em um primeiro momento o estudo dos modelos neurais despertou grande interesse, mas este sofreu uma redução significativa, em parte devido ao trabalho de Minsky e Papert [33] que tratava dos limites fundamentais que um perceptron de camada única poderia computar. O estudo das redes neurais artificiais continuou de forma reduzida até a

década de 80.

O desenvolvimento da *segunda geração* de neurônios artificiais ocorreu na década de 80 [16] e permitiu que os modelos tivessem saídas contínuas, múltiplas camadas e pudessem ser treinadas com algoritmos baseados no gradiente descendente, como o backpropagation [59]. Estas redes se mostraram bastante eficientes e permitiram a realização de diversas aplicações práticas como, por exemplo, reconhecimento de padrões em sistemas de reconhecimento de odores [61, 60], diagnósticos de doenças [26], reconhecimento de caracteres manuscritos [14]; aproximação de funções e previsão de séries temporais como, por exemplo, previsões climáticas [24], financeiras [29, 42], entre outras [18, 50]. Apesar do desenvolvimento e contínuo estudo das redes neurais artificiais, os modelos baseados em neurônios de *segunda geração* ainda apresentam diversos problemas, como lentidão na convergência, sobre ajuste de dados, problemas de generalização e convergência em mínimos locais. Uma das abordagens para tentar resolver estes problemas é a utilização de modelos com maior inspiração biológica [28].

Foi verificado experimentalmente que os neurônios biológicos apresentam saídas em forma de pulsos e não como números escalares. Desta forma surgiram as redes neurais artificiais *spiking*, que possuem como saída um pulso e não um escalar [46]. Denominada a *terceira geração* de redes neurais têm tido maior ênfase nas pesquisas atuais e acredita-se que poderão resolver alguns dos problemas encontrados nos modelos de *segunda geração*.

As redes neurais artificiais podem ser consideradas como um modelo não convencional de computação, pois não é baseada no modelo de von Neumann. Outro modelo não convencional surgiu com as idéias da mecânica quântica aplicadas à computação, denominada computação quântica, proposta por Feynman em 1982 [13]. Acredita-se que este novo modelo de computação trará vantagens computacionais frente à computação clássica. Isto se deve principalmente ao fato da descoberta do algoritmo de busca de Grover [15] e do algoritmo quântico para a fatoração em tempo polinomial [49], que pode ser utilizado para quebrar o sistema de criptografia RSA [49]. Apesar da existência de algoritmos quânticos para fatoração e busca que são mais eficientes do que os algoritmos clássicos existentes, estes não podem ser utilizados, pois o computador quântico ainda está em desenvolvimento.

Da união destas linhas de pesquisa foi criado um novo paradigma. As redes neurais artificiais quânticas foram inicialmente propostas por Kak em 1995 [19]. Ainda não se sabe se o cérebro utiliza propriedades da mecânica quântica em seu processamento, isto faria com que a simulação do processamento cerebral fosse possível apenas em um computador quântico. Nos trabalhos de Penrose [43] argumenta-se que a computação quântica é parte fundamental para

simulação do cérebro, mas em outro trabalho Tegmark argumenta que o cérebro provavelmente não utiliza propriedades quânticas [54]. Apesar de não existir a certeza de que as redes neurais biológicas utilizam propriedades inerentemente quânticas em seu funcionamento, acredita-se que as redes neurais artificiais quânticas trarão vantagens computacionais frente às redes neurais artificiais clássicas assim como a computação quântica apresenta vantagens sobre a sua contra parte clássica [49, 15]. Isto pode ser observado nos diversos trabalhos que citam a utilização da computação quântica para o desenvolvimento de modelos neurais [19, 6, 23, 32, 21].

Mesmo após o trabalho de Kak [19] existiam poucas pesquisas nesta área, pois as redes neurais artificiais utilizam funções não lineares que dificilmente terão análogas quânticas exatas. Apesar disto, Altaisky [6] mostrou que uma regra de aprendizado para perceptrons pode ser utilizada em sistemas quânticos. Atualmente, diversas redes neurais quânticas (RNQ) já foram propostas, mas a maioria apresenta problemas quanto aos postulados da mecânica quântica ou aos princípios da computação neural.

Neste trabalho serão utilizados modelos de redes neurais sem peso (RNSP) que não utilizam funções de ativação não lineares. As RNSP não possuem pesos ajustáveis associados às suas conexões, suas entradas e saídas são binárias e as funções dos nodos são armazenadas em tabelas verdades (posições de memória) que podem ser armazenadas em memórias de acesso aleatório (RAM).

Uma das características marcantes das RNSP é sua facilidade de treinamento, que consiste basicamente em alterar valores de endereçamentos de memória. Outra vantagem é que podem ser implementadas diretamente em *hardware* clássico [8]. Buscamos manter estas vantagens construindo nosso modelo de forma que ele fosse diretamente implementado em hardware quântico, os circuitos quânticos, e que o treinamento fosse realizado através da manipulação de qubits em determinados registros, que chamamos de seletores. Em seguida, será mostrado que o algoritmo de treinamento clássico para redes piramidais de Igor Aleksander pode ser diretamente adaptado para as qRNSP [39], o que não ocorre nos modelos de RNQ com peso. Pretende-se, então, unir o poder da computação quântica às redes neurais sem peso para gerar redes poderosas e fáceis de treinar.

A programação de computadores quânticos não é uma tarefa simples e não é possível a construção de um computador quântico programável e determinístico [35]. Logo as redes neurais quânticas, e os outros modelos de aprendizagem de máquina, podem vir a ser uma das principais metodologias para resolução de problemas em um computador quântico.

1.2 Objetivos

O objetivo principal desta dissertação é verificar se técnicas da computação quântica podem ser aplicadas com sucesso no desenvolvimento de redes neurais artificiais para a tarefa de reconhecimento de padrões. Onde o sucesso no desenvolvimento significa que os modelos desenvolvidos devem ser mais eficientes, viáveis, robustos, preferencialmente que apresentem vantagens computacionais quando comparados a sua contraparte clássica e que sejam de fácil operação, não necessitando de alteração nos circuitos quânticos para o seu funcionamento, sem violar os princípios da Mecânica Quântica.

O primeiro passo dado para a realização deste objetivo foi realizado em 2008 [39], onde os modelos PLN e MPLN foram generalizados aos domínios da computação quântica de forma que as propriedades clássicas do modelo foram preservadas. A estrutura dos modelos foi alterada para que algoritmos quânticos possam ser implementados utilizando os nodos lógicos quânticos.

Entre os objetivos específicos desta dissertação estão:

1. Definir um modelo de rede neural quântica sem pesos com as seguintes características:
i) implementação direta em circuitos quânticos; ii) capacidade de simular os algoritmos clássicos; iii) Estrutura que permita a utilização de algoritmos de aprendizado quânticos;
2. Definir um algoritmo de aprendizado supervisionado para redes neurais sem peso quânticas onde todos os padrões do conjunto de treinamento sejam apresentados a rede simultaneamente (em superposição). Este algoritmo pode ser desenvolvido utilizando, por exemplo, um algoritmo de busca quântico ou uma memória associativa quântica;
3. Definir um modelo de rede neural sem peso que possua vantagem exponencial em relação ao tamanho da memória. Isto será realizado utilizando como base para definição do neurônio uma memória quântica probabilística [55], assim como as redes neurais sem peso foram desenvolvidas baseadas na memória RAM.
4. Verificar se os modelos de redes neurais quânticas com pesos respeitam os postulados da mecânica quântica e os princípios da computação neural.

1.3 Organização da dissertação

A principal dificuldade para compreender os postulados da Mecânica Quântica, não são os postulados em si, e sim a grande quantidade de conceitos de álgebra linear requeridos para compreendê-los, junto com a notação de Dirac utilizada pelos físicos [36]. No Capítulo 2, será realizada uma breve revisão dos conceitos de álgebra linear utilizando a notação de Dirac.

No capítulo 3 são descritos os conceitos sobre computação quântica [36, 17] necessários para o desenvolvimento do trabalho. A partir dos postulados da mecânica quântica são derivados os conceitos sobre bits quânticos e suas operações. As vantagens computacionais da computação quântica é demonstrada através do protocolo de teletransporte da informação quântica e de uma análise do algoritmo de busca de Groover.

No capítulo 4 é descrita uma memória quântica probabilística que tem a capacidade de armazenar 2^n padrões utilizando apenas n bits quânticos, a memória quântica probabilística será utilizada para definir um modelo de rede neural denominado nodo PQM.

No capítulo 5 são descritos os modelos de redes neurais sem peso utilizados nesta dissertação. As técnicas abordadas são as redes formadas por nodos RAM, nodos lógicos probabilísticos (PLN), nodos lógicos probabilísticos multi-valorados (MPLN) e o goal seeking neuron (GSN). O funcionamento das redes sem peso é descrito detalhadamente, de forma que uma visão geral sobre os modelos seja obtida.

Os modelos de redes neurais quânticas baseados no neurônio MCP serão descritos no capítulo 6. Será demonstrado que os modelos de RNQ com pesos apresentados possuem problemas quanto aos postulados da mecânica quântica, este é um dos fatores que apontam em direção à pesquisa sobre as redes neurais quânticas sem peso.

No capítulo 7 são descritos os modelos de redes neurais quânticas sem peso definidos nesta dissertação, denominados nodo lógico probabilístico quântico (qPLN), nodo lógico probabilístico multi-valorado quântico (qMPLN), nodo RAM quântico (qRAM) e o nodo baseado na memória quântica probabilística (PQM). Estes modelos foram desenvolvidos de forma a respeitar os postulados da mecânica quântica e permitir que os algoritmos clássicos fossem diretamente adaptados. Em seguida é apresentado um algoritmo de treinamento para redes neurais sem peso utilizando os princípios da computação quântica onde os padrões do conjunto de treinamento são apresentados a rede simultaneamente.

CAPÍTULO 2

Álgebra Linear

A compreensão dos postulados da mecânica quântica para representações discretas requer diversos conceitos de álgebra linear. A representação dos vetores na literatura da área de computação quântica é realizada através da notação de Dirac [36]. Neste capítulo será realizada uma breve revisão dos conceitos de álgebra linear utilizando a notação de Dirac.

Definição 1. *Um espaço vetorial consiste de:*

1. *Um corpo F de escalares;*
2. *Um conjunto V de objetos denominados vetores;*
3. *Uma regra, dita adição de vetores, que associa a cada par de vetores α, β em V um vetor $\alpha + \beta$ em V , denominado a soma de α e β , de maneira tal que:*
 - *A adição é comutativa, $\alpha + \beta = \beta + \alpha$;*
 - *A adição é associativa, $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$;*
 - *Existe um único vetor 0 em V , denominado o vetor nulo, tal que $\alpha + 0 = \alpha$ para todo α em V ;*
 - *Para cada vetor α em V existe um único vetor $-\alpha$ em V tal que $\alpha + (-\alpha) = 0$*
4. *Uma regra, dita multiplicação escalar, que associa a cada escalar c em F e cada vetor α em V um vetor $c\alpha$ em V , denominado o produto de c por α de maneira tal que:*
 - *Para todo elemento α de V $1\alpha = \alpha$ para todo α em V ;*
 - *Se c_1, c_2 são números, então $(c_1 c_2)\alpha = c_1(c_2\alpha)$;*
 - *Se c é um número, então $c(\alpha + \beta) = c\alpha + c\beta$;*
 - *Se c_1, c_2 são números, então $(c_1 + c_2)\alpha = c_1\alpha + c_2\alpha$.*

Para este trabalho o espaço vetorial de maior interesse será C^n , o espaço de todas as n -uplas de números complexos (z_1, z_2, \dots, z_n) . A notação padrão para um vetor em um espaço vetorial na mecânica quântica é a descrita na equação 2.1,

$$|\psi\rangle \quad (2.1)$$

onde ψ é um rótulo para o vetor. A notação $|\cdot\rangle$ é utilizada para indicar que o objeto é um vetor. O objeto inteiro $|\psi\rangle$ é chamado de *ket*.

2.1 Conjuntos Geradores, Independência Linear e Bases

Definição 2. Um subconjunto não vazio W de um espaço vetorial V é um subespaço de V se, e somente se, para cada par de vetores $|\psi\rangle, |\phi\rangle$ em W e cada escalar c em F , o vetor $c|\psi\rangle + |\phi\rangle$ está em W

Definição 3. Seja S um conjunto de vetores num espaço vetorial V . O subespaço gerado por S é definido como sendo a interseção de todos os subespaços de V que contém S . Quando S é um conjunto finito de vetores, $S = \{|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_n\rangle\}$, denominaremos W simplesmente como o subespaço gerado pelos vetores $|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_n\rangle$.

Teorema 1. O subespaço gerado por um subconjunto não vazio S de um espaço vetorial V é o conjunto de todas as combinações lineares de vetores em S .

Prova: Seja W o subespaço gerado por S . Então, cada combinação linear

$$|\psi\rangle = a_1 |\psi_1\rangle + a_2 |\psi_2\rangle + \dots + a_m |\psi_m\rangle$$

de vetores $|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_n\rangle$ em S evidentemente está em W . Assim W contém o conjunto L de todas as combinações lineares de vetores em S . O conjunto L , por outro lado, contém S e é não vazio. Se $|\psi\rangle, |\phi\rangle$ pertencem a L , então $|\psi\rangle$ é uma combinação linear,

$$|\psi\rangle = a_1 |\psi_1\rangle + a_2 |\psi_2\rangle + \dots + a_n |\psi_n\rangle$$

de vetores $|\psi_i\rangle$ em S e $|\phi\rangle$ é uma combinação linear

$$|\phi\rangle = b_1 |\phi_1\rangle + b_2 |\phi_2\rangle + \dots + b_n |\phi_n\rangle$$

de vetores $|\varphi_j\rangle$ em S . Para cada escalar c ,

$$c|\psi\rangle + |\varphi\rangle = \sum_{i=1}^m (c \cdot a_i) |\psi_i\rangle + \sum_{j=1}^n (c \cdot b_j) |\psi_j\rangle$$

Logo, $c|\psi\rangle + |\varphi\rangle$ pertence a L . Assim, L é um subespaço de V . \square

Mostramos acima que L é um subespaço de V que contém S e também que todo subespaço que contém S contém L . Decorre que L é a interseção de todos os subespaços que contém S , isto é, que L é o subespaço gerado pelo conjunto S .

Alguns autores preferem definir um subconjunto de espaço vetorial gerado por um conjunto S como todas as combinações lineares de vetores em S .

Definição 4. Um conjunto de vetores $|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle$ de um espaço vetorial é chamado *linearmente dependente* se existe um conjunto de escalares a_1, a_2, \dots, a_n de um corpo F com pelo menos um $a_i \neq 0$ tal que

$$\sum_{i=1}^n a_i |v_i\rangle = 0$$

Um conjunto de vetores é *linearmente independente* se não for linearmente dependente.

Definição 5. Seja V um espaço vetorial. Uma *base* de V é um conjunto linearmente independente de vetores em V que gera o espaço V . O espaço vetorial V é de *dimensão finita* se ele possui uma base finita.

2.2 Operadores Lineares

Definição 6. Um *operador linear* (ou *transformação linear*) entre espaços vetoriais V e W é uma aplicação $A : V \rightarrow W$ que é linear em suas entradas:

$$A\left(\sum_i a_i |v_i\rangle\right) = \sum_i a_i A(|v_i\rangle)$$

Usualmente escreve-se $A|v\rangle$ para denotar $A(|v\rangle)$. Diz-se que A é um operador linear definido sobre um espaço V quando A é uma função de V em V .

O conceito de aplicação é muito geral, se V e W são dois espaços vetoriais não nulos, então

existem muitas aplicações de V em W , mas as transformações lineares podem ser definidas apenas pela ação nos vetores de uma base de V .

Teorema 2. *Seja V um espaço vetorial de dimensão finita e seja $\{|v_1\rangle, \dots, |v_n\rangle\}$ uma base ordenada de V . Seja W um espaço vetorial e $\{|w_1\rangle, \dots, |w_n\rangle\}$ vetores arbitrários em W . Então existe uma única transformação linear T de V em W tal que*

$$T|v_j\rangle = |w_j\rangle, j = 1, \dots, n$$

Prova: Dado um vetor $|v\rangle$ em V , existe uma única n-upla (a_1, \dots, a_n) tal que

$$|v\rangle = a_1|v_1\rangle + \dots + a_n|v_n\rangle$$

Definimos, para este vetor,

$$T|v\rangle = a_1|w_1\rangle + \dots + a_n|w_n\rangle$$

É evidente que $T|v_j\rangle = |w_j\rangle$. Para ver que T é linear, seja

$$|w\rangle = b_1|v_1\rangle + \dots + b_n|v_n\rangle$$

em V e c um escalar arbitrário.

$$c|v\rangle + |w\rangle = (ca_1 + b_1)|v_1\rangle + \dots + (ca_n + b_n)|v_n\rangle$$

portanto

$$T(c|v\rangle + |w\rangle) = (ca_1 + b_1)|w_1\rangle + \dots + (ca_n + b_n)|w_n\rangle$$

Por outro lado

$$c(T|v\rangle) + T|w\rangle = c \sum_{i=1}^n a_i |w_i\rangle + \sum_{i=1}^n b_i |w_i\rangle = \sum_{i=1}^n (ca_i + b_i) |w_i\rangle$$

Logo,

$$T(c|v\rangle + |w\rangle) = c(T|v\rangle) + T|w\rangle$$

Se U é uma transformação linear de V em W com $U|v_j\rangle = |w_j\rangle, j = 1, \dots, n$, então, para o vetor $|v\rangle = \sum_{i=1}^n a_i |v_i\rangle$, temos

$$U|v\rangle = U \left(\sum_{i=1}^n a_i |v_i\rangle \right) = \sum_{i=1}^n a_i (U|v_i\rangle) = \sum_{i=1}^n a_i |w_i\rangle$$

De modo que U é exatamente a regra T definida acima. Portanto T é única. \square

2.3 Representação de Transformações Lineares por Matrizes

A matriz de uma transformação linear é um objeto concreto, associado a essa transformação na presença de bases ordenadas em seu domínio e contra domínio [25].

Seja $A : V \rightarrow W$ uma transformação linear entre os espaços V e W . Suponha que $|v_1\rangle, \dots, |v_m\rangle$ formam uma base finita de V e que $|w_1\rangle, \dots, |w_n\rangle$ formam uma base finita de W . Então, para cada j ($j = 1, \dots, m$), existem números complexos A_{1j}, \dots, A_{nj} tais que

$$A|v_j\rangle = \sum_i A_{ij}|w_i\rangle$$

A matriz cujas entradas são os valores A_{ij} é a representação matricial do operador A em relação as bases ordenadas $\{|v_i\rangle\}$ e $\{|w_j\rangle\}$.

Além disso, se A é uma $m \times n$ matriz arbitrária, então podemos definir uma transformação linear, pois

$$A \left(\sum_i a_i |v_i\rangle \right) = \sum_i a_i A |v_i\rangle$$

A equação acima se verifica, pois o produto de matrizes é associativo. A equação define uma transformação cuja representação matricial é A .

Podemos resumir formalmente da seguinte forma:

Teorema 3. *Seja V um espaço vetorial n -dimensional e W um espaço vetorial m -dimensional. Seja $\{|v_i\rangle\}$ uma base ordenada finita de V e $\{|w_j\rangle\}$ uma base ordenada de W . Para cada transformação linear T de V em W , existe uma matriz A $m \times n$ tal que*

$$T|v\rangle = A|v\rangle,$$

para todo vetor $|v\rangle$ em V . Além disso $T \rightarrow A$ é uma correspondência bijetora entre o conjunto das transformações lineares de V em W e o conjunto das $m \times n$ matrizes sobre os complexos.

Pode-se demonstrar que se A é uma transformação linear de um espaço vetorial V em um espaço vetorial W , e B é uma transformação linear do espaço vetorial W em um espaço vetorial

X . Então a representação matricial da transformação BA é uma matriz igual ao produto das representações matriciais de A e B , com respeito às bases apropriadas.

2.4 Produto Interno

O conceito de produto interno enriquece a noção de espaço vetorial, permitindo uma linguagem geométrica e o destaque de certas transformações lineares especiais [25].

Definição 7. Uma função (\cdot, \cdot) de $V \times V$ em C é um *produto interno* se satisfaz as seguintes propriedades:

1. (\cdot, \cdot) é linear no segundo argumento;
2. $(|v\rangle, |w\rangle) = (|w\rangle, |v\rangle)^*$;
3. $(|v\rangle, |v\rangle) \geq 0$, com a igualdade ocorrendo se, e somente se, $|v\rangle = 0$.

A notação padrão para o produto interno $(|v\rangle, |w\rangle)$ na mecânica quântica é $\langle v|w\rangle$, onde $|v\rangle$ e $|w\rangle$ são vetores no espaço vetorial e $\langle v|$ é o dual conjugado do vetor $|v\rangle$.

Exemplo 1. C^n possui um produto interno definido por $((y_1, \dots, y_n), (z_1, \dots, z_n)) = \sum_i y_i^* z_i$

Textos sobre mecânica quântica geralmente tratam sobre *espaços de Hilbert* no espaço vetorial complexo finito. “Em espaços vetoriais complexos com dimensão finita utilizados na computação quântica, um espaço de Hilbert é exatamente o mesmo que um espaço com produto interno” [36].

Definição 8. Um conjunto de vetores $\{|v_k\rangle\}$ são ditos *ortogonais*, se para $i \neq j$ temos que $\langle v_i|v_j\rangle = 0$.

Definição 9. A *norma* de um vetor $|v\rangle$ é dada por $\sqrt{\langle v|v\rangle}$

Um vetor $|v\rangle$ é dito *normal* se $\langle v|v\rangle = 1$, isto é $\| |v\rangle \| = 1$

Definição 10. Um conjunto de vetores $\{|v_k\rangle\}$ são ditos *ortonormais*, se são normais e ortogonais simultaneamente.

No teorema abaixo, mostraremos que todo espaço vetorial finito com produto interno, possui uma base ortogonal. O processo de construção desta base é conhecido como procedimento de Gram-Schmidt.

Teorema 4. *Seja V um espaço com produto interno e sejam $|w_1\rangle, |w_2\rangle, \dots, |w_n\rangle$ vetores independentes arbitrários em V . Então, é possível construir vetores ortonormais $|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle$ tais que $\{|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle\}$ é uma base do espaço gerado por $|w_1\rangle, |w_2\rangle, \dots, |w_n\rangle$*

Demonstração. Defina $|v_1\rangle = |w_1\rangle / |||w_1\rangle||$ e para $1 \leq k \leq n-1$ defina $|v_{k+1}\rangle$ indutivamente por

$$|v_{k+1}\rangle \equiv \frac{|w_{k+1}\rangle - \sum_{i=1}^k \langle v_i | w_{k+1} \rangle |v_i\rangle}{|||w_{k+1}\rangle - \sum_{i=1}^k \langle v_i | w_{k+1} \rangle |v_i\rangle||}$$

2.5 Autovalores e Autovetores

Definição 11. *Seja V um espaço vetorial e seja T um operador linear sobre V . Um autovalor de T é um escalar c em F tal que existe um vetor não nulo $|v\rangle$ em V com $T|v\rangle = c|v\rangle$. Se c é um autovalor de T , então*

1. *todo $|v\rangle$ tal que $T|v\rangle = c|v\rangle$ é chamado autovetor de T associado ao autovalor c ;*
2. *a coleção de todos os $|v\rangle$ tais que $T|v\rangle = c|v\rangle$ é denominada espaço característico associado a c .*

2.6 Adjunta e Operadores Hermitianos

Seja A um operador linear sobre um espaço vetorial complexo finito V com produto interno. Existe um único operador linear A^\dagger tal que para todos vetores $|v\rangle, |w\rangle \in V$,

$$(|v\rangle, A|w\rangle) = (A^\dagger|v\rangle, |w\rangle)$$

Este operador é conhecido como o Adjunto do operador A e sua matriz associada como Adjunta. Facilmente podemos perceber que $(AB)^\dagger = B^\dagger A^\dagger$, pois

$$(|v\rangle, AB|w\rangle) = (|v\rangle, A(B|w\rangle)) = (A^\dagger|v\rangle, B|w\rangle) = (B^\dagger A^\dagger|v\rangle, |w\rangle).$$

Definimos também $|v\rangle^\dagger = \langle v|$. Um operador A cuja adjunta $A^\dagger = A$ é chamado de *auto-adjunto* ou *Hermitiano*.

Definição 12. Um operador linear U é dito *unitário* se $U^\dagger U = I$

Operadores unitários são importantes por preservarem o produto interno, pois

$$(U|v\rangle, U|w\rangle) = (|v\rangle, U^\dagger U|w\rangle) = (|v\rangle, |w\rangle).$$

2.7 Produto Tensorial

O produto tensorial é utilizado para ‘juntar’ espaços vetoriais, formando espaços vetoriais de maior dimensão.

Sejam V e W espaços vetoriais de dimensão m e n , respectivamente. Então $V \otimes W$ (leia-se V tensorial W) terá dimensão $m \cdot n$. Os elementos de $V \otimes W$ serão produtos tensoriais $|v\rangle \otimes |w\rangle$, com $|v\rangle$ em V e $|w\rangle$ em W . Se $|i\rangle$ e $|j\rangle$ são bases ortonormais de V e W , então $|i\rangle \otimes |j\rangle$ é uma base do espaço vetorial $V \otimes W$. Podemos denotar $|v\rangle \otimes |w\rangle$ utilizando as notações abreviadas $|v\rangle |w\rangle$, $|v, w\rangle$, $|vw\rangle$.

Propriedades básicas do produto tensorial:

1. Para um escalar arbitrário z e elementos $|v\rangle$ de V e $|w\rangle$ de W , temos que

$$z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$$

2. Para arbitrários $|v_1\rangle$ e $|v_2\rangle$ em V e $|w\rangle$ em W ,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle;$$

3. Para $|v\rangle$ arbitrário em V e $|w_1\rangle, |w_2\rangle$ em W ,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle.$$

Definição 13. Sejam $|v\rangle$ e $|w\rangle$ vetores nos espaços vetoriais V e W , respectivamente, e A e B operadores lineares sobre V e W definimos $A \otimes B$ pela seguinte equação:

$$(A \otimes B)(|v\rangle \otimes |w\rangle) = A|v\rangle \otimes B|w\rangle$$

O produto interno nos espaços V e W podem ser usados para definir um produto interno natural em $V \otimes W$, dado por:

$$\left(\sum_i a_i |v_i\rangle \otimes |w_i\rangle, \sum_j a_j |v'_j\rangle \otimes |w'_j\rangle \right) = \sum_{ij} a_i^* b_j \langle v_i | v'_j \rangle \langle w_i | w'_j \rangle.$$

Em espaços finitamente dimensionais podemos realizar toda discussão acima em termos matriciais. Seja A uma matriz $m \times n$ e B uma matriz $p \times q$. Então:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}$$

Exemplo 2. $\begin{bmatrix} 7 \\ 9 \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \cdot 3 \\ 7 \cdot 6 \\ 9 \cdot 3 \\ 9 \cdot 6 \end{bmatrix} = \begin{bmatrix} 21 \\ 42 \\ 27 \\ 54 \end{bmatrix}.$

Computação Quântica

Neste capítulo é realizada uma introdução aos conceitos da computação quântica necessários para o entendimento desta dissertação. Na seção 3.2 são apresentados os postulados da mecânica quântica que servem como base para a computação quântica. Na seção 3.3 é definido o bit quântico, ou qubit. Na seção 3.4 são definidos os operadores sobre qubits. Na seção 3.5 a medição de qubits é descrita. Na seção 3.6 a representação de múltiplos qubits é descrita. Na seção 3.7 é apresentado o paralelismo quântico, que é uma das propriedades mais importantes da computação quântica. Na seção 3.8 verifica-se que um qubit não pode ser copiado. Este resultado é conhecido como o teorema da não clonagem. Na seção 3.9 são descritos os circuitos quânticos e na seção 3.10 são descritos o algoritmo de busca quântico e o algoritmo de Deutsch.

3.1 Introdução

A Computação Quântica [36] foi originalmente proposta por Richard Feynman [13], que propôs que a simulação de sistemas quânticos por um computador clássico só poderia ser realizada com um custo exponencial. Por outro lado, um computador quântico pode simular um sistema quântico em tempo polinomial. Dessa forma, os computadores quânticos seriam mais eficientes do que os computadores clássicos na simulação de sistemas quânticos.

Uma formalização para a computação quântica foi proposta por David Deutsch [12] (que propôs um modelo universal de computação baseado nos princípios da mecânica quântica: a máquina de Turing quântica). A computação quântica é um domínio de pesquisa recente que utiliza elementos de três áreas bem conhecidas: Matemática, Física e Computação. A computação quântica permite o desenvolvimento de algoritmos que superam os algoritmos clássicos conhecidos. Por exemplo, o algoritmo de fatoração em tempo polinomial [49], que possui ganho exponencial ao melhor algoritmo clássico conhecido e o algoritmo de busca que

tem um ganho quadrático sobre o melhor algoritmo clássico [15].

3.2 Postulados da Mecânica Quântica

Nesta seção são descritos os quatro postulados da mecânica quântica para dois instantes de tempo como apresentados em [36]. Uma descrição dos postulados considerando tempo contínuo pode ser encontrada em [47]. É importante lembrar que a mecânica quântica é uma teoria em desenvolvimento, mas nesta dissertação será considerado que os postulados estão completos, e que a construção de um computador quântico é possível. Os postulados serão utilizados nas seções 3.3 a 3.8 para definir os conceitos da computação quântica sob um aspecto matemático.

Postulado 1. *Qualquer sistema físico isolado está associado a um espaço vetorial complexo V com produto interno (um espaço de Hilbert). Um sistema é completamente descrito por um vetor unitário (vetor de estado) em V .*

Postulado 2. *A evolução de um sistema quântico fechado é descrito por uma transformação unitária. Isto é, o estado $|\psi\rangle$ de um sistema no instante t_1 está relacionado com o estado $|\psi'\rangle$ do sistema no instante t_2 por uma transformação unitária U que depende apenas de t_1 e t_2 ,*

$$|\psi'\rangle = U |\psi\rangle$$

Postulado 3. *Medições quânticas são descritas por uma coleção $\{M_m\}$ de operadores de medição. Estes operadores agem sobre o espaço vetorial associado ao sistema que está sendo medido. O índice m refere-se às possíveis saídas que podem ocorrer na medição. Se o estado do sistema quântico é $|\psi\rangle$ imediatamente antes da medição ocorrer, então a probabilidade do resultado m ocorrer é dada por*

$$p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle.$$

e o estado do sistema após a medida será:

$$\frac{M_m |\psi\rangle}{\sqrt{\langle \psi | M_m^\dagger M_m | \psi \rangle}}.$$

Os operadores de medida devem satisfazer a relação de completitude:

$$\sum_m M_m^\dagger M_m = I$$

Postulado 4. O estado de um sistema físico composto é o produto tensorial dos estados dos sistemas físicos componentes. Além disso, se os sistemas estiverem numerados de 1 a n , e o sistema número i estiver no estado $|\psi_i\rangle$, então o estado do sistema total é $|\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$.

3.3 Qubits

A computação (denominada neste trabalho *computação clássica*) é construída utilizando como conceito fundamental o bit, de forma análoga a computação quântica é construída utilizando como conceito fundamental o bit quântico ou qubit. Um qubit pode ser representado fisicamente pelo spin de um elétron [22] ou pelos diferentes estados de polarização da luz [6]. Um qubit representa um sistema físico, mas neste trabalho um qubit será definido como um objeto matemático abstrato que obedece a algumas propriedades, como visto no Postulado 1. Este ponto de vista permitirá que seja realizada uma análise do problema de interesse independente do sistema físico que um computador quântico venha a ser construído [36].

Definição 14. Um **bit quântico** (ou **qubit**) $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, é um vetor complexo bidimensional unitário pertencente ao espaço vetorial C^2 , onde α e β são números complexos, $\{|0\rangle, |1\rangle\}$ é uma base ortonormal de C^2 . Os vetores $|0\rangle$ e $|1\rangle$ são denominados os estados da **base computacional**.

A principal diferença entre os bits e os qubits é que os qubits podem estar em superposição (combinação linear) de estados. Seja $|\psi\rangle$ um qubit, então podemos escrever:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (3.1)$$

onde $|\alpha|^2 + |\beta|^2 = 1$ e

$$|0\rangle \leftrightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle \leftrightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Enquanto um bit clássico pode estar em apenas dois estados 0 ou 1, um qubit pode estar em um espaço contínuo de estados. Dessa forma, em um único qubit poderia ser armazenada uma quantidade infinita de informação. No entanto a observação de um qubit não revela suas amplitudes α e β . Na computação clássica a medição não é considerada, pois não possui efeito significativo no processo computacional. Na computação quântica uma medição interfere no sistema, como será visto na seção 3.5. Podemos examinar um bit clássico para descobrir se este se encontra no estado 0 ou 1, mas não podemos determinar o estado quântico de um qubit. Da medição de um qubit é obtida apenas uma informação bem mais restrita. Obtem-se $|0\rangle$ com probabilidade $|\alpha|^2$ ou $|1\rangle$ com probabilidade $|\beta|^2$. O bit quântico poderá se encontrar simultaneamente nos estados $|0\rangle$ e $|1\rangle$, mas após medição este irá colapsar, por exemplo, para um dos estados da base computacional.

Apesar da dimensionalidade de um qubit, existe uma representação geométrica conhecida como esfera de Bloch. Como $|\alpha|^2 + |\beta|^2 = 1$, podemos reescrever a equação 3.1 como

$$\begin{aligned} |\psi\rangle &= \alpha|0\rangle + \beta|1\rangle = e^{i\gamma}|\alpha||0\rangle + e^{i\zeta}|\beta||1\rangle \\ &= e^{i\gamma}(|\alpha||0\rangle + e^{i\varphi}|\beta||1\rangle), \end{aligned}$$

onde $e^{i\gamma}$ não possui efeitos observáveis, então podemos escrever

$$|\psi\rangle = |\alpha||0\rangle + e^{i\varphi}|\beta||1\rangle$$

e como $|\alpha|^2 + |\beta|^2 = 1$ então existe $\frac{\theta}{2}$ tal que

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle. \quad (3.2)$$

Os números θ e φ definem um ponto em uma esfera de raio unitário, como mostrado na figura 3.1. Esta esfera, chamada esfera de Bloch, provê uma visualização de um estado quântico de um único qubit. No entanto, esta intuição é limitada, pois não existe representação gráfica para múltiplos qubits.

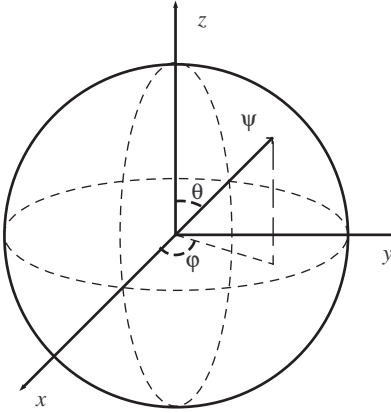


Figura 3.1 Representação de um qubit na Esfera de Bloch

3.4 Operações sobre qubits

A definição de operadores quânticos é baseada no postulado 2. Desta forma um operador quântico sobre um único qubit $A \rightarrow C^2 \times C^2$ é definido como um operador linear e unitário em C^2 . Apesar de não estar explicito no postulado qualquer operador linear unitário sobre C^2 define um operador quântico sobre um qubit.

Alguns exemplos de operações quânticas são os operadores identidade **I**, que mantém o estado do qubit inalterado; e a operação Hadamard **H**, que ‘cria’ superposição de estados.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Um conjunto de operadores quânticos conhecidos como matrizes de Pauli é formado pelo operador identidade e pelos operadores definidos na equação 3.3.

$$\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3.3)$$

Para ilustrar a ação de um operador quântico sobre um qubit, considere a ação do operador

X sobre os estados da base computacional, descrita nas equações 3.4 e 3.5.

$$\mathbf{X}|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \quad (3.4)$$

$$\mathbf{X}|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \quad (3.5)$$

Após verificar a ação de um operador na base computacional sua ação em um estado em superposição fica definida pela linearidade dos operadores, por exemplo na equação 3.6 o operador **X** é aplicado a um estado quântico $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

$$\mathbf{X}|\psi\rangle = \mathbf{X}(\alpha|0\rangle + \beta|1\rangle) = \mathbf{X}\alpha|0\rangle + \mathbf{X}\beta|1\rangle = \alpha|1\rangle + \beta|0\rangle = \beta|0\rangle + \alpha|1\rangle \quad (3.6)$$

Lema 1. *Seja x um número real e A uma matriz tal que $A^2 = I$, então $\exp(iAx) = \cos(x)I + i \cdot \sin(x)A$*

Demonstração: Para verificar este fato, lembre que $\exp(x)$ pode ser escrito como um série infinita $\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$, logo

$$\exp(iAx) = 1 + (iAx) + \frac{(iAx)^2}{2!} + \frac{(iAx)^3}{3!} + \frac{(iAx)^4}{4!} + \dots,$$

agrupando separadamente os termos com expoentes pares e os termos com expoentes ímpares se obtém

$$\begin{aligned} \exp(iAx) &= \left(1 + \frac{(iAx)^2}{2!} + \frac{(iAx)^4}{4!} + \dots\right) + \left((iAx) + \frac{(iAx)^3}{3!} + \dots\right) \\ &= \sum_{i=0}^{\infty} \frac{(iAx)^{2i}}{2i!} + \sum_{i=0}^{\infty} \frac{(iAx)^{2i+1}}{(2i+1)!} = \sum_{i=0}^{\infty} \frac{(ix)^{2i}}{2i!} + \sum_{i=0}^{\infty} \frac{(iAx)^{2i+1}}{(2i+1)!} \text{ pois } A^2 = I \end{aligned}$$

Note que o primeiro somatório é a expansão em uma série infinita do cosseno e o segundo a expansão do seno, portanto conclui-se como queríamos demonstrar que

$$\exp(iAx) = \cos(x)I + i \cdot \sin(x)A$$

□

Facilmente pode se verificar que o quadrado das matrizes de Pauli resultam na identidade.

Definimos uma rotação de ângulo $\theta \in \mathbb{R}$ sobre os eixos x, y ou z , denotadas respectivamente por $R_x(\theta)$, $R_y(\theta)$ e $R_z(\theta)$ como descrito nas equações 3.7, 3.8 e 3.9.

$$R_x(\theta) = \exp(-i\theta X/2) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad (3.7)$$

$$R_y(\theta) = \exp(-i\theta Y/2) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad (3.8)$$

$$R_z(\theta) = \exp(-i\theta Z/2) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \quad (3.9)$$

Os operadores de rotação podem ser generalizados para serem utilizados em qualquer direção.

Definição 15. Se $v = (v_1, v_2, v_3)$ é um vetor real unitário, então a operação

$$R_v(\theta) = \exp\left(\frac{-i\theta v \vec{\sigma}}{2}\right) = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)(v_1\mathbf{X} + v_2\mathbf{Y} + v_3\mathbf{Z})$$

onde $\vec{\sigma} = (X, Y, Z)$, é uma operação de rotação com ângulo θ sobre o eixo v .

A definição 15 é útil, pois permite expressar qualquer operador sobre um único qubit como uma operação de rotação seguida de uma multiplicação por uma fase global.

3.5 Medição

Na seção 3.3 é descrito o procedimento para a medição de um qubit, mas aparentemente não existe uma relação entre o procedimento descrito e o Postulado 3, que descreve a medição de sistemas quânticos.

Nesta seção mostraremos que a medição descrita na seção 3.3 é um caso particular do Postulado 3. Para isto, considere os operadores 3.10.

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \text{ e } M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.10)$$

Com os operadores 3.10 a probabilidade de obter 0 como resultado da medição do qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ é descrita na equação 3.11 e o estado resultante desta medição é descrito na equação 3.13. A probabilidade de obter 1 é descrita na equação 3.12 e o estado resultante na equação 3.14.

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = \langle\psi|\alpha|0\rangle = \alpha\alpha^*\langle 0|0\rangle + \beta\alpha^*\langle 1|0\rangle = |\alpha|^2 \quad (3.11)$$

$$p(1) = \langle\psi|M_1^\dagger M_1|\psi\rangle = \langle\psi|M_1|\psi\rangle = \langle\psi|\beta|1\rangle = \alpha\beta^*\langle 0|1\rangle + \beta\beta^*\langle 1|1\rangle = |\beta|^2 \quad (3.12)$$

$$|\psi'\rangle = \frac{M_0|\psi\rangle}{\sqrt{\langle\psi|M_0^\dagger M_0|\psi\rangle}} = \frac{\alpha|0\rangle}{|\alpha|} \approx |0\rangle \quad (3.13)$$

$$|\psi'\rangle = \frac{M_1|\psi\rangle}{\sqrt{\langle\psi|M_1^\dagger M_1|\psi\rangle}} = \frac{\beta|1\rangle}{|\beta|} \approx |1\rangle \quad (3.14)$$

A medição definida pelos operadores M_0 e M_1 é conhecida como medida projetiva e será o único tipo de medição utilizado nesta dissertação.

3.6 Múltiplos Qubits

Para representar sistemas com múltiplos qubits é utilizado o produto tensorial. Suponha que tenhamos 2 qubits, então teremos 4 estados da base computacional $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$. Um sistema de 2 qubits também pode estar em superposição e o quadrado da soma de suas amplitudes deverá ter norma 1.

Em um sistema de dois qubits $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$ podemos medir, por exemplo o primeiro qubit, sendo $|\alpha_{00}|^2 + |\alpha_{01}|^2$ a probabilidade de obtermos $|0\rangle$ como

resposta e neste caso, o estado do sistema pós medição seria

$$|\psi'\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}.$$

De forma mais geral se $|\psi\rangle$ representa um sistem quântico com n qubits, então o estado de $|\psi\rangle$ pode ser descrito pela equação 3.15, com 2^n amplitudes, onde $\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$.

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle \quad (3.15)$$

3.7 Paralelismo

Uma das propriedades mais importantes da computação quântica é o paralelismo quântico. Seja U_f um operador quântico sobre dois qubits que implementa uma função $f : B \rightarrow B$, tal que $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$. Aplicando o operador U_f a um estado em superposição $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$, obtem-se por linearidade o estado descrito na equação 3.16.

$$\begin{aligned} U_f \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \right) &= \frac{1}{\sqrt{2}}(U_f|00\rangle + U_f|10\rangle) \\ &= \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle) \end{aligned} \quad (3.16)$$

O operador U_f foi aplicado ao estado $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$ e o estado resultante 3.16 possui o valor de $f(0)$ e de $f(1)$. Esse fenômeno em que o valor de $f(x)$ é "avaliado" simultaneamente para dois valores de x é denominado *paralelismo quântico*.

O paralelismo quântico pode ser generalizado para funções $f : B^n \rightarrow B^n$. Para isto é necessário criar o estado descrito na equação 3.17 e aplicar o operador U_f obtendo o estado descrito na equação 3.18 onde o valor de $f(x)$ é calculado para todo $x \in B^n$ simultaneamente.

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle \quad (3.17)$$

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x\rangle |f(x)\rangle \quad (3.18)$$

O estado $\sum_{x=0}^{n-1} |x\rangle$ pode ser obtido facilmente, para isto basta aplicar o operador Hadamard sobre n qubits inicializados no estado $|0\rangle$. Por exemplo, na equação 3.19 a criação do estado $\sum_{x=0}^{n-1} |x\rangle$ é exemplificada com $n = 2$.

$$\begin{aligned} H^{\otimes 2} |00\rangle &= H \otimes H |0\rangle \otimes |0\rangle = H |0\rangle \otimes H |0\rangle \\ &= \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \end{aligned} \quad (3.19)$$

O estado 3.18 não possui uma aplicação imediata, pois se observarmos o estado 3.18 será obtida apenas uma de suas parcelas com igual probabilidade. Na seção 3.10.1 será mostrado um exemplo de aplicação do paralelismo quântico.

3.8 Não clonagem

Nesta seção será apresentado um resultado que mostra que estados quânticos não podem ser copiados, este fato é conhecido como o teorema da não clonagem de estados quânticos.

Teorema 5. *Não clonagem: Bits quânticos não podem ser copiados*

Demonstração. A prova será realizada por contradição. Suponha que existe uma máquina U que copie estados quânticos. Sejam $|a_1\rangle$ e $|a_2\rangle$ dois estados ortogonais e que $|a_1\rangle$ é o estado em "branco". Então $U(|a_1\rangle |a_1\rangle) = |a_1\rangle |a_1\rangle$ e $U(|a_2\rangle |a_1\rangle) = |a_2\rangle |a_2\rangle$. Considere o estado $\frac{1}{\sqrt{2}}(|a_1\rangle + |a_2\rangle)$,

$$\begin{aligned} U \left(\frac{1}{\sqrt{2}} (|a_1\rangle + |a_2\rangle) |a_1\rangle \right) &= \left(\frac{1}{\sqrt{2}} (|a_1\rangle + |a_2\rangle) \right) \left(\frac{1}{\sqrt{2}} (|a_1\rangle + |a_2\rangle) \right) = \\ &= \frac{1}{2} (|a_1\rangle |a_1\rangle + |a_1\rangle |a_2\rangle + |a_2\rangle |a_1\rangle + |a_2\rangle |a_2\rangle) \end{aligned} \quad (3.20)$$

mas pela linearidade de U

$$\begin{aligned} \mathbf{U} \left(\frac{1}{\sqrt{2}} (|a_1\rangle + |a_2\rangle) |a_1\rangle \right) &= \frac{1}{\sqrt{2}} \mathbf{U}(|a_1\rangle |a_1\rangle) + \frac{1}{\sqrt{2}} \mathbf{U}(|a_2\rangle |a_1\rangle) = \\ &= \frac{1}{\sqrt{2}} |a_1\rangle |a_1\rangle + \frac{1}{\sqrt{2}} |a_2\rangle |a_2\rangle \end{aligned} \quad (3.21)$$

Os valores obtidos para $\mathbf{U} \left(\frac{1}{\sqrt{2}} (|a_1\rangle + |a_2\rangle) |a_1\rangle \right)$ nas equações (3.20) e (3.21) são diferentes, o que é uma contradição. \square

3.9 Circuitos Quânticos

A linguagem de circuitos é a mais utilizada na descrição dos algoritmos quânticos. Um circuito quântico é análogo a um circuito booleano formado, por exemplo, por portas E e NÃO. Na Figura 3.2 é descrito o circuito do operador Hadamard sobre o qubit $|0\rangle$ ($H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$).

$$|0\rangle \longrightarrow \boxed{H} \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Figura 3.2 Circuito operador Hadamard

Na Figura 3.3 é mostrado o circuito que representa uma medição. Esta operação projeta um estado quântico em superposição a um estado quântico da base computacional (um bit clássico). A linha dupla no circuito distingue serve para assinalar que este é um bit. Se $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ então a saída do circuito descrito na Figura 3.3 será 0 com probabilidade $|\alpha|^2$ e será 1 com probabilidade $|\beta|^2$.

$$|\psi\rangle \longrightarrow \boxed{\text{Medição}} \longrightarrow \text{Saída}$$

Figura 3.3 Símbolo circuito de medição

O circuito do operador denominado não controlado (C_{NOT}) é descrito na figura 3.4. Uma representação alternativa para o circuito do operador não controlado é descrito na Figura 3.5.

O operador não controlado pode ser utilizado para copiar estados da base computacional. A representação matricial do operador C_{NOT} é dada na equação (3.22).

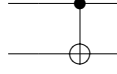


Figura 3.4 Circuito operador C_{NOT}

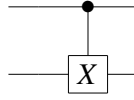


Figura 3.5 Outra representação circuito operador C_{NOT}

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.22)$$

3.9.1 Teleporte quântico

Com os conceitos vistos até o momento pode ser descrito o circuito de teleporte de informação quântica. Na comunidade de criptografia existe uma tradição de chamar duas partes que estão se comunicando pelos nomes Alice e Bob.

Suponha que Alice possui um qubit no estado descrito na equação 3.23.

$$\alpha |0\rangle + \beta |1\rangle \quad (3.23)$$

E que as amplitudes α e β do estado 3.23 são desconhecidas. Suponha também que Alice deseja enviar este estado para Bob.

Uma possibilidade é enviar o qubit para Bob através de um canal de informação quântico, mas suponha que Alice pode enviar apenas informação clássica para Bob. Esta tarefa parece

ser impossível, pois Alice não conhece as amplitudes do estado (3.23), logo não pode enviar instruções de construção do estado 3.23 para Bob.

Alice poderia medir o estado (3.23) e enviar para Bob o resultado da medição, 0 ou 1. Esta tentativa irá falhar se α ou β forem diferentes de 0. Mesmo que infinitas cópias do estado (3.23) estivessem disponíveis para Alice a única informação que ela poderia estimar seriam os valores de $|\alpha|^2$ e $|\beta|^2$. Mesmo que Alice obtivesse os valores exatos de $|\alpha|^2$ e $|\beta|^2$ a informação não seria suficiente para que Bob reconstruísse o qubit, pois Bob não poderia decidir, por exemplo, se o estado 3.23 é $\alpha|0\rangle + \beta|1\rangle$ ou $\alpha|0\rangle - \beta|1\rangle$.

É impossível para Alice enviar o bit quântico para Bob utilizando apenas um canal clássico. No entanto, a tarefa pode ser realizada se Alice e Bob compartilharem um par EPR.

Suponha que Alice e Bob possuem dois qubits em um estado conhecido como par EPR, descrito na equação 3.24. Este estado pode ser obtido através do circuito descrito na figura 3.6.

$$|\beta_{00}\rangle \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \quad (3.24)$$

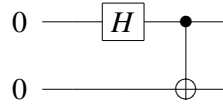


Figura 3.6 Circuito para construção de pares EPR

Então o estado inicial dos qubits utilizados no protocolo de teletransporte é descrito na equação 3.25. Onde os dois qubits mais a esquerda são o qubits de Alice e o qubit a direita é o qubit de Bob.

$$\begin{aligned} & (a|0\rangle + b|1\rangle) \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ &= \frac{a}{\sqrt{2}}|0\rangle|00\rangle + \frac{a}{\sqrt{2}}|0\rangle|11\rangle + \frac{b}{\sqrt{2}}|1\rangle|00\rangle + \frac{b}{\sqrt{2}}|1\rangle|11\rangle \\ &= \frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|100\rangle + \frac{b}{\sqrt{2}}|111\rangle \end{aligned} \quad (3.25)$$

1. Alice aplica o operador não controlado em seus qubits utilizando o mais a esquerda como

controle. O estado 3.25 irá se tornar o estado descrito na equação 3.26

$$\frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|110\rangle + \frac{b}{\sqrt{2}}|101\rangle \quad (3.26)$$

2. A próxima ação de Alice é aplicar a transformação Hadamard no qubit mais a esquerda. O resultado é descrito na equação 3.27.

$$\begin{aligned} & \frac{a}{\sqrt{2}} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |00\rangle + \frac{a}{\sqrt{2}} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |11\rangle + \\ & \frac{b}{\sqrt{2}} \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) |10\rangle + \frac{b}{\sqrt{2}} \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) |01\rangle \end{aligned} \quad (3.27)$$

A estado 3.27 pode ser reescrito como 3.28

$$\begin{aligned} & \frac{a}{2}|000\rangle + \frac{a}{2}|100\rangle + \frac{a}{2}|011\rangle + \frac{a}{2}|111\rangle \\ & + \frac{b}{2}|010\rangle + \frac{b}{2}|110\rangle + \frac{b}{2}|001\rangle + \frac{b}{2}|101\rangle \end{aligned} \quad (3.28)$$

E separando o qubit de Bob 3.28 pode ser reescrito como 3.29.

$$\begin{aligned} & \frac{1}{2}|00\rangle a|0\rangle + \frac{1}{2}|10\rangle a|0\rangle + \frac{1}{2}|01\rangle a|1\rangle + \frac{1}{2}|11\rangle a|1\rangle \\ & + \frac{1}{2}|01\rangle b|0\rangle + \frac{1}{2}|11\rangle b|0\rangle + \frac{1}{2}|001\rangle b|1\rangle + \frac{1}{2}|10\rangle b|1\rangle \end{aligned} \quad (3.29)$$

E finalmente, 3.29 pode ser reescrito como 3.30.

$$\begin{aligned} & \frac{1}{2}|00\rangle (a|0\rangle + b|1\rangle) + \frac{1}{2}|01\rangle (a|1\rangle + b|0\rangle) \\ & \frac{1}{2}|10\rangle (a|0\rangle - b|1\rangle) + \frac{1}{2}|11\rangle (a|1\rangle - b|0\rangle) \end{aligned} \quad (3.30)$$

3. Agora Alice observa seus dois qubits. O estado resultante irá depender do valor observado por Alice, como descrito na Tabela 3.1.

4. Alice envia para Bob o resultado de sua observação (dois bits clássicos)

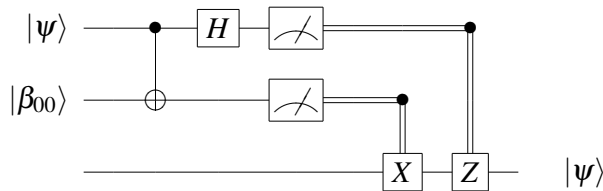
5. A ação de Bob depende dos bits enviados por Alice

observação de Alice	Estado após observação
00	$ 00\rangle (a 0\rangle + b 1\rangle)$
01	$ 01\rangle (a 1\rangle + b 0\rangle)$
10	$ 10\rangle (a 0\rangle - b 1\rangle)$
11	$ 11\rangle (a 1\rangle - b 0\rangle)$

Tabela 3.1

- Se os bits de Alice forem 00, Bob não faz nada. Seu qubit já está no estado $\alpha|0\rangle + \beta|1\rangle$.
- Se os bits de Alice forem 01, Bob aplica o operador X em seu qubit, obtendo o estado $\alpha|0\rangle + \beta|1\rangle$.
- Se os bits de Alice forem 10, Bob aplica o operador Z em seu qubit, obtendo o estado $\alpha|0\rangle + \beta|1\rangle$.
- Se os bits de Alice forem 11, Bob aplica primeiro o operador X e depois o operador Z , obtendo novamente o estado $\alpha|0\rangle + \beta|1\rangle$.

A figura 3.7 descreve o protocolo do teleporte quântico, onde os dois qubits superiores estão com Alice e o qubit inferior está com Bob. No circuito as linhas simples representam qubits e as linhas duplas representam bits clássicos.

**Figura 3.7** Circuito quântico para teleportar um qubit

Experimentos realizando o teleporte quântico foram relatados em Note que o estado de Alice foi destruído após a execução do protocolo, por isso não houve uma cópia da informação. Também é importante salientar que o teleporte não transporta o qubit e sim o estado do qubit.

3.10 Algoritmos Quânticos

Nesta seção serão descritos algoritmos quânticos que apresentam custo computacional inferior a qualquer algoritmo clássico conhecido para resolução de um dado problema.

3.10.1 Algoritmo de Deutsch

Seja f uma função $f : B \rightarrow B$, o algoritmo de Deutsch é utilizado para calcular o valor de $f(0) \oplus f(1)$ com uma única avaliação da função f . O algoritmo de Deutsch é útil para mostrar o uso do paralelismo para obter uma informação global de $f(x)$. O algoritmo de Deutsch é mais rápido que qualquer estratégia clássica, pois no caso clássico teríamos necessariamente que calcular os valores de $f(0)$ e $f(1)$ para obtermos $f(0) \oplus f(1)$.

Algoritmo 1: Algoritmo de Deutsch

Entrada: U_f

- 1 Inicialize $|\psi_0\rangle = |01\rangle$
 - 2 $|\psi_1\rangle = H^{\otimes 2} |\psi_0\rangle$
 - 3 $|\psi_2\rangle = U_f |\psi_1\rangle$
 - 4 $|\psi_3\rangle = H \otimes I |\psi_2\rangle$
 - 5 Realize uma medição do primeiro qubit para obter o valor de $f(0) \oplus f(1)$
-

O algoritmo 1 descreve o algoritmo de Deutsch. No passo 1 os registradores quânticos são inicializados com o estado $|01\rangle$ e obtem-se $|\psi_0\rangle = |01\rangle$. No passo 2 o operador $H^{\otimes 2}$ é utilizado para criar uma superposição de estados,

$$|\psi_1\rangle = H^{\otimes 2} |\psi_0\rangle = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right] \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right].$$

A ação do operador U_f sobre um estado $|x\rangle (|0\rangle - |1\rangle) / \sqrt{2}$ levará ao estado $(-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) / \sqrt{2}$. Este fato pode ser facilmente verificado. Se $f(x) = 0$, então

$$U_f \left(|x\rangle (|0\rangle - |1\rangle) / \sqrt{2} \right) = \frac{1}{\sqrt{2}} (U_f |x\rangle |0\rangle - U_f |x\rangle |1\rangle) = \frac{1}{\sqrt{2}} (|x\rangle |0\rangle - |x\rangle |1\rangle) =$$

$$|x\rangle (|0\rangle - |1\rangle) / \sqrt{2} = (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) / \sqrt{2}.$$

E se $f(x) = 1$, então

$$\begin{aligned} U_f \left(|x\rangle (|0\rangle - |1\rangle) / \sqrt{2} \right) &= \frac{1}{\sqrt{2}} (U_f |x\rangle |0\rangle - U_f |x\rangle |1\rangle) = \frac{1}{\sqrt{2}} (|x\rangle |1\rangle - |x\rangle |0\rangle) = \\ &= -|x\rangle (|0\rangle - |1\rangle) / \sqrt{2} = (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) / \sqrt{2}. \end{aligned}$$

No passo 3 do algoritmo o valor $U_f |\psi_1\rangle$ e o estado $|\psi_2\rangle$ estará em um dos quatro possíveis estados descritos na equação 3.31.

$$\begin{cases} |\psi_2\rangle = \pm \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right] \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] & \text{se } f(0) = f(1) \\ |\psi_2\rangle = \pm \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] & \text{se } f(0) \neq f(1) \end{cases} \quad (3.31)$$

No passo 4 do algoritmo 1 é aplicado o operador $H \otimes I$ sobre o estado $|\psi_2\rangle$ e obtem-se o estado $|\psi_3\rangle$ descrito na equação 3.32.

$$\begin{cases} |\psi_3\rangle = \pm |0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] & \text{se } f(0) = f(1) \\ |\psi_3\rangle = \pm |1\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] & \text{se } f(0) \neq f(1) \end{cases} \quad (3.32)$$

Como $f(0) \oplus f(1) = 0$ se $f(0) = f(1)$ e $f(0) \oplus f(1) = 1$ se $f(0) \neq f(1)$, a equação 3.32 pode ser reescrita como na equação 3.33.

$$|\psi_3\rangle = \pm |f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \quad (3.33)$$

A medição do primeiro qubit no passo 5 irá retornar o valor de $f(0) \oplus f(1)$ com apenas uma chamada do operador U_f .

O algoritmo de Deutsch pode ser representado por um circuito quântico, como na Figura 3.8.

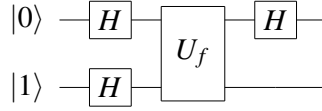


Figura 3.8 Circuito quântico do algoritmo de Deutsch

3.10.2 Algoritmo de Busca

Em 1996 L.K. Grover propôs um algoritmo quântico de busca sobre dados não ordenados que é mais rápido que qualquer algoritmo clássico conhecido. Dada uma base de dados com N itens o algoritmo clássico mais eficiente irá necessitar de uma média de $0,5N$ passos clássicos antes de encontrar o item desejado. No pior caso serão necessários N passos clássicos. O algoritmo de Grover supera qualquer algoritmo clássico e realiza a tarefa em $O(\sqrt{N})$ passos quânticos.

O algoritmo 2 descreve a busca quântica, neste a busca é realizada em um espaço com N elementos. Considerando que $N = 2^n$ então o conjunto dos índices I podem ser armazenados em n bits, a busca pode ser realizada apenas sobre os índices de cada elemento, que são números entre 0 e $N - 1$ e o problema de busca pode ter M soluções, com $1 \leq M \leq N$.

Um problema de busca pode ser caracterizado por uma função $f : \{0, 1\}^n \rightarrow \{0, 1\}$ onde se $x_0 \in \{0, 1\}^n$ é o valor procurado, então a função f assume os valores descritos em 3.34.

$$f(x) = \begin{cases} 1, & \text{se } x = x_0 \\ 0, & \text{se } x \neq x_0 \end{cases} \quad (3.34)$$

A partir da função 3.34 pode ser definido um operador unitário U_f tal que $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$. Uma primeira tentativa para solucionar o problema é inicializar o estado $|x\rangle$ com todas as possíveis instâncias do problema e $|y\rangle = 0$ e aplicar o operador U_f , explorando o princípio da superposição. Na equação 3.35 as operações para um problema n -dimensional são descritas matricialmente.

$$U_f ((H^{\otimes n} \otimes I) |\mathbf{0}, 0\rangle) = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}, f(\mathbf{x})\rangle \quad (3.35)$$

Realizando uma medição dos qubits no registrador \mathbf{x} no estado 3.35 será obtido um dos 2^n

strings em $\{0, 1\}^n$ com igual probabilidade. Se uma medição for realizada no registrador \mathbf{y} será obtido 0 com probabilidade $\frac{2^n-1}{2^n}$ e $|1\rangle$ com probabilidade $\frac{1}{2^n}$. Se o resultado da medição for $|1\rangle$ o registrador \mathbf{x} irá conter a resposta correta, mas isto irá ocorrer com uma probabilidade muito baixa.

Para aumentar a probabilidade de obter a resposta desejada ao medir o registrador \mathbf{y} do estado 3.35 o algoritmo de Grover utiliza dois truques: *inversão de fase* e a *inversão sobre a média*.

A inversão de fase muda a fase do estado desejado. Para isto basta inicializar o registrador \mathbf{y} com o estado $\frac{|0\rangle - |1\rangle}{\sqrt{2}} = H|1\rangle$. Ao aplicar o operador U_f com o registrador \mathbf{x} inicializado com um estado da base computacional e o registrador \mathbf{y} com o estado $H|1\rangle$, obtem-se o valor descrito na equação 3.36

$$\begin{aligned} U_f(I_n \otimes H|\mathbf{x}, 1\rangle) &= U_f\left(|\mathbf{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) \\ &= |\mathbf{x}\rangle \left(\frac{|0 \oplus f(\mathbf{x})\rangle - |1 \oplus f(\mathbf{x})\rangle}{\sqrt{2}}\right) = |\mathbf{x}\rangle \left(\frac{|f(\mathbf{x})\rangle - |\overline{f(\mathbf{x})}\rangle}{\sqrt{2}}\right) \\ &= (-1)^{f(\mathbf{x})} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \begin{cases} -1 |\mathbf{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) & \text{se } \mathbf{x} = \mathbf{x}_0 \\ +1 |\mathbf{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) & \text{se } \mathbf{x} \neq \mathbf{x}_0 \end{cases} \end{aligned} \quad (3.36)$$

Se $|\mathbf{x}\rangle$ for inicializado em uma superposição de todos os possíveis strings com n bits, então a inversão de fase irá marcar com um sinal negativo o estado desejado. Por exemplo, se o string procurado for '01', então após realizar a inversão de fase, o estado do qubit $|\mathbf{x}\rangle$ será descrito pelo vetor $[\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$. A amplitude do estado $|01\rangle$ será marcada com um sinal negativo, i.e., com uma alteração de fase. A inversão de fase marcou o estado desejado, mas medir $|\mathbf{x}\rangle$ não trará nenhuma informação, pois $|\frac{1}{2}|^2 = |-\frac{1}{2}|^2$.

Para aumentar a amplitude de probabilidade do estado marcado com uma mudança de fase é utilizado um procedimento denominado inversão sobre a média. Para ilustrar o procedimento de inversão sobre a média considere os números 30, 34, 79, 7 e 75. A média desses números é $a = 45$. O procedimento de inversão da média consiste em alterar os números que estão acima da média para um valor abaixo da média preservando a distância entre a média e o número. E alterar os números que estão abaixo da média para um valor acima da média, também mantendo a distância do número a média. O procedimento consiste em inverter cada elemento em relação a média. Para o primeiro número temos $a - 30 = 15$ unidades de distância

da média. Adicionando $a = 45$ a 15 obtem-se $a + (a - 30) = 50$. A nova sequência será 60, 56, 11, 83 e 15. Note que a média da nova sequência permanece 45. Em geral cada elemento v será alterado como na equação 3.37.

$$v' = v + 2a \quad (3.37)$$

O procedimento de inversão da média pode ser escrito matricialmente. Escrevendo a sequência como um vetor $V = (30, 34, 79, 7, 75)^T$, podemos calcular a média dos valores com a matriz A descrita na equação 3.38. Pode ser verificado facilmente que $AV = (45, 45, 45, 45, 45)^T$.

$$A = \begin{pmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{pmatrix} \quad (3.38)$$

A equação 3.37 pode ser reescrita com em 3.39.

$$V' = -V + 2AV = (-I + 2A)V \quad (3.39)$$

Generalizando a matriz A para 2^n entradas o operador $(-I + 2A)$ é descrito na equação 3.40

$$-I + 2A = \begin{pmatrix} -1 + \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \frac{1}{2^n} & -1 + \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2^n} & \frac{1}{2^n} & \cdots & -1 + \frac{1}{2^n} \end{pmatrix} \quad (3.40)$$

Para verificar que a ação combinada da inversão de fase com a inversão sobre a média considere o seguinte vetor $V = (5, 5, 5, 5, 5)$. Realizando uma inversão de fase no segundo número obtem-se $V = (5, -5, 5, 5, 5)$. A média desses números é 3. Calculando a inversão sobre a média obtem-se

$$-v + 2a = -5 + (2 \cdot 3) = 1 \quad (3.41)$$

$$-v + 2a = 5 + (2 \cdot 3) = 11 \quad (3.42)$$

E a nova sequência será $V = (1, 11, 1, 1, 1)$.

O algoritmo de Grover é descrito no Algoritmo 2. O Algoritmo 2 é baseado na inversão de fase e inversão sobre a média utilizados nos passos 3 e 4 que são conhecidos como iteração de Grover. A iteração de Grover deve ser repetida 2^n vezes e em seguida uma medição dos qubits pode ser realizada.

Algoritmo 2: Algoritmo de busca quântico

- 1 Inicialize o sistema $|\psi\rangle = |0\rangle_n$
 - 2 Aplique o operador Hadamard $|\psi\rangle = H^{\otimes n} |\psi\rangle$
 - 3 Aplique a operação de inversão de fase $U_f(I \otimes H)$
 - 4 Aplique a inversão sobre a média $-I + 2A$
 - 5 Repita os passos 3 e 4 $\sqrt{2^n}$ vezes
 - 6 Realiza uma medição dos qubits
-

Memória quântica probabilística

As vantagens da computação quântica normalmente estão relacionadas com a melhoria no desempenho de algoritmos, como por exemplo, o algoritmo de fatoração de Shor [49] e o algoritmo de busca de Grover [15]. Neste capítulo será descrita uma memória probabilística quântica QPM, onde a capacidade de armazenamento é incrementada significativamente, para armazenar 2^n padrões com n bits quânticos (qubits) serão utilizados apenas n qubits. Na seção 4.2 é descrito o algoritmo de armazenamento da informação em uma memória QPM, na seção 4.3 é descrito o algoritmo para recuperar informações de uma memória QPM, o processo de recuperação é probabilístico e a seção 4.4 é um sumário do capítulo.

4.1 Introdução

A maioria das operações quânticas necessárias neste capítulo foram detalhadas na seção operadores quânticos, no entanto são introduzidos dois operadores específicos para este capítulo S^i e U .

$$S^i = \begin{pmatrix} \sqrt{\frac{i-1}{i}} & \frac{1}{\sqrt{i}} \\ -\frac{1}{\sqrt{i}} & \sqrt{\frac{i-1}{i}} \end{pmatrix} \quad U = \begin{pmatrix} \exp(i\frac{\pi}{2n}) & 0 \\ 0 & 1 \end{pmatrix} \quad (4.1)$$

A idéia para o armazenamento dos padrões é que dados p padrões p^i seja criado um estado quântico em que os padrões p^i estejam em superposição, como na equação (4.2). Para criar este estado de superposição dos padrões pode ser utilizado o algoritmo proposto por Trugenberger em [56], descrito na na seção 4.2 e para o processo de recuperação de informação será utilizado o algoritmo descrito na seção 4.3.

$$|m\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^p |p^i\rangle \quad (4.2)$$

4.2 Armazenamento

Para construir o estado $|m\rangle$ será necessário inicializar um qubit $|\psi_0^1\rangle$ (equação (4.3)) com três registros: um registro \mathbf{p} com n qubits, onde os padrões serão inseridos iterativamente no sistema; um registro auxiliar \mathbf{u} com 2 qubits; e um registro \mathbf{m} com n qubits, inicializado com o valor $|0\rangle_n$, onde serão armazenados os qubits com a memória $|m\rangle$.

A idéia do algoritmo de armazenamento é separar este estado em 2 termos, uma correspondente aos padrões armazenados e a outra pronta para o processo de inserção de novos padrões. O estado do segundo qubit auxiliar u_2 distingue estas duas partes, onde $u_2 = |0\rangle$ para os padrões armazenados e $u_2 = |1\rangle$ para o termo em processamento.

$$|\psi_0^1\rangle = |p_1^1, \dots, p_n^1; 01; 0_1, \dots, 0_n\rangle \quad (4.3)$$

Para cada padrão p^i devem ser realizadas as seguintes operações:

$$|\psi_1^i\rangle = \prod_{j=1}^n 2XOR_{p_j^i u_2 m_j} |\psi_0^i\rangle \quad (4.4)$$

A ação do operador descrito na equação (4.4) é a de copiar o conteúdo do registro p para o registro de memória \mathbf{m} .

$$|\psi_2^i\rangle = \prod_{j=1}^n NOT_{m_j} XOR_{p_j^i m_j} |\psi_1^i\rangle \quad (4.5)$$

O operador descrito na equação (4.5) faz com que todos os qubits do registro de memória \mathbf{m} assumam o valor $|1\rangle$ quando o conteúdo do padrão p^i e do registro de memória são idênticos, isto ocorre exatamente para o termo em processamento.

$$|\psi_3^i\rangle = nXOR_{m_1 \dots m_n u_1} |\psi_2^i\rangle \quad (4.6)$$

Em seguida através da operação descrita na equação (4.6) o qubit auxiliar u_1 do termo em processamento tem seu valor alterado para $|1\rangle$.

$$|\psi_4^i\rangle = CS_{u_1 u_2}^{p+1-i} |\psi_3^i\rangle \quad (4.7)$$

A equação (4.7) descreve o operador que tem papel central no algoritmo de armazenamento, ela tem o papel de separar o novo padrão a ser armazenado com a devida normalização.

$$|\psi_5^i\rangle = nXOR_{m_1 \dots m_n u_1} |\psi_4^i\rangle \quad (4.8)$$

$$|\psi_6^i\rangle = \prod_{j=n}^1 XOR_{p^j m_j} NOT_{m_j} |\psi_5^i\rangle \quad (4.9)$$

As equações (4.8) e (4.9) descrevem operadores que são inversos aos descritos nas equações (4.6) e (4.5), têm como finalidade restaurar o qubit auxiliar u_1 e o registro de memória \mathbf{m} para seus valores iniciais. Neste momento o estado dos registros é descrito pela equação (4.10).

$$|\psi_6^i\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^i |p^i; 00; p^k\rangle + \sqrt{\frac{p-i}{p}} |p^i; 01; p^i\rangle \quad (4.10)$$

A última operação, descrita na equação (4.11), restaura o registro \mathbf{m} do termo em processamento (o segundo fator da equação (4.10)) para seu valor inicial $|0_1, \dots, 0_n\rangle$. Nesse momento um novo padrão pode ser carregado no registro p e toda a rotina deve ser repetida. No final do processo o registro \mathbf{m} irá conter o estado $|m\rangle$. Uma iteração do processo é descrita através de um circuito quântico na figura 4.1.

$$|\psi_7^i\rangle = \prod_{j=n}^1 2XOR_{p^j u_2 m_j} |\psi_6^i\rangle \quad (4.11)$$

Para uma melhor compreensão do algoritmo será ilustrado o seu funcionamento para armazenar os padrões $p^1 = |00\rangle$ e $p^2 = |01\rangle$. Inicialmente é preparado o estado $|\psi_0^1\rangle = |p^1; 01; 00\rangle = |00; 01; 00\rangle$, no próximo passo tem-se $|\psi_1^1\rangle = |\psi_0^1\rangle$, pois $p^1 = |00\rangle$. Aplicando a operação descrita na equação (4.5) obtem-se $|\psi_2^1\rangle = |00; 01; 11\rangle$ e em seguida $|\psi_3^1\rangle = |00; 11; 11\rangle$. A ação do operador S^2 faz com que o estado entre em superposição, $|\psi_4^1\rangle = \frac{1}{\sqrt{2}} (|00; 10; 11\rangle + |00; 11; 11\rangle)$ e em seguida são aplicados os operadores descritos nas equações (4.8), (4.9) e (4.11), que restauram o valor de u_1 e do registrador \mathbf{m} , $|\psi_5^1\rangle = \frac{1}{\sqrt{2}} (|00; 00; 11\rangle + |00; 01; 11\rangle)$ e $|\psi_6^1\rangle = \frac{1}{\sqrt{2}} (|00; 00; 00\rangle + |00; 01; 00\rangle) = |\psi_7^1\rangle$.

Na próxima iteração do algoritmo a utilidade do registro auxiliar u poderá ser percebida. Cria-se o estado $|\psi_0^2\rangle$ a partir do estado $|\psi_7^1\rangle$ alimentando no registro p o padrão p^2 , logo

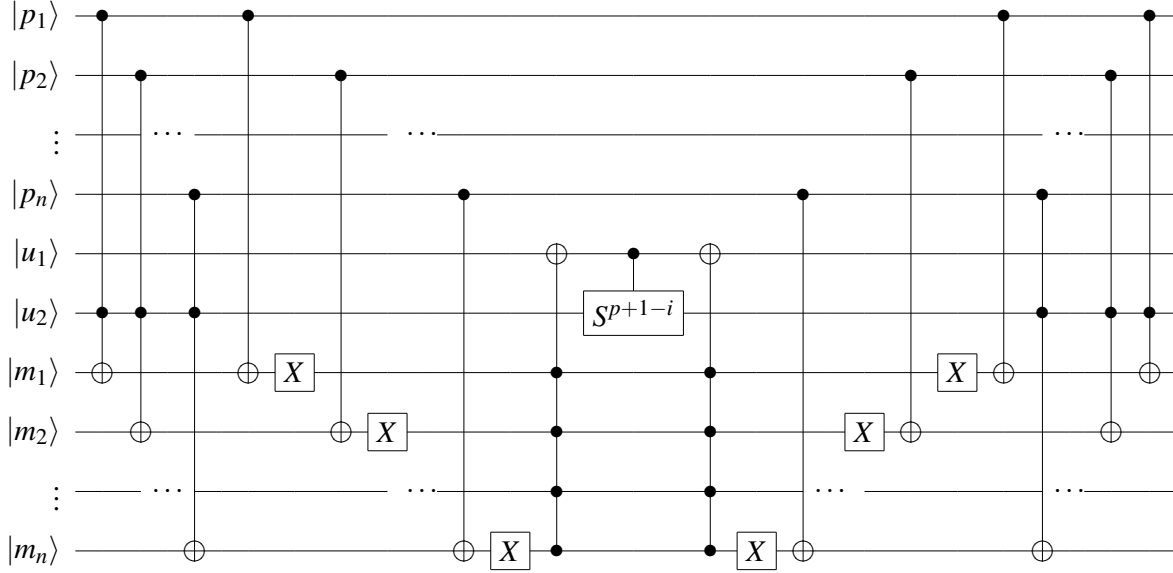


Figura 4.1 Circuito armazenamento memória probabilística quântica

$|\psi_0^2\rangle = \frac{1}{\sqrt{2}}(|01;00;00\rangle + |01;01;00\rangle)$, note que u_2 servirá para marcar que fator está em processamento, obtem-se no próximo passo

$$|\psi_1^2\rangle = \frac{1}{\sqrt{2}}(|01;00;00\rangle + |01;01;01\rangle), \quad (4.12)$$

desta forma o padrão p^2 será copiado apenas no registro de memória do fator a direita na equação (4.12), o que fará com que o valor de u_1 seja alterado apenas neste fator $|\psi_2^2\rangle = \frac{1}{\sqrt{2}}(|01;00;10\rangle + |01;01;11\rangle)$ e $|\psi_3^2\rangle = \frac{1}{\sqrt{2}}(|01;00;10\rangle + |01;11;11\rangle)$, o que irá determinar a ação do operador S , chegando então ao estado $|\psi_4^2\rangle = \frac{1}{\sqrt{2}}(|01;00;10\rangle + |01;10;11\rangle)$, em seguida são realizadas as operações descritas nas equações (4.8), (4.9) e (4.11), obtendo-se respectivamente $|\psi_5^2\rangle = \frac{1}{\sqrt{2}}(|01;00;10\rangle + |01;00;11\rangle)$, $|\psi_6^2\rangle = \frac{1}{\sqrt{2}}(|01;00;00\rangle + |01;00;01\rangle)$ e finalmente o estado $|\psi_7^2\rangle = \frac{1}{\sqrt{2}}[|0100\rangle(|00\rangle + |01\rangle)]$, onde no registro de memória \mathbf{m} está armazenado o estado $|m\rangle$.

4.3 Recuperação

O algoritmo de recuperação também requer a inicialização de um estado quântico com três registros. O primeiro registro **i** com n qubits irá conter o padrão de entrada $|i\rangle$; o segundo registro **m**, também com n qubits, armazena a memória $|m\rangle$ e o terceiro registro **c** contém apenas um qubit de controle $|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, o estado inicial $|\psi_0\rangle$ é descrito na equação (4.13).

$$|\psi_0\rangle = \frac{1}{\sqrt{2p}} \sum_{k=1}^p |i_1, \dots, i_n; p_1^k; \dots; p_n^k; 0\rangle + \frac{1}{\sqrt{2p}} \sum_{k=1}^p |i_1, \dots, i_n; p_1^k; \dots; p_n^k; 1\rangle \quad (4.13)$$

Após a inicialização são aplicadas as seguintes combinações de operadores quânticos:

$$|\psi_1\rangle = \prod \text{NOT}_{m_k} \text{XOR}_{i_k m_k} |\psi_0\rangle \quad (4.14)$$

A equação (4.14) que faz com que os qubits no registro de memória **m** fiquem no estado $|1\rangle$ se $i_j = p_j^k$ e $|0\rangle$ caso contrário. O estado $|\psi_1\rangle$ é descrito na equação (4.15), onde $d_i^k = 1$ se e somente se $i_j = p_j^k$ e $d_j^k = 0$ caso contrário.

$$|\psi_1\rangle = \frac{1}{\sqrt{2p}} \sum_{k=1}^p |i_1, \dots, i_n; d_1^k; \dots; d_n^k; 0\rangle + \frac{1}{\sqrt{2p}} \sum_{k=1}^p |i_1, \dots, i_n; d_1^k; \dots; d_n^k; 1\rangle \quad (4.15)$$

O valor de $|\psi_2\rangle$ é calculado através do operador descrito na equação (4.16), onde $\mathcal{H} = (d\mathcal{H})_m \otimes (\sigma_3)_c$ e $(d\mathcal{H})_m = \sum_{k=1}^n \frac{\sigma_3 + 1}{2}_{m_k}$.

$$|\psi_2\rangle = \exp\left(i \frac{\pi}{2n} \mathcal{H}\right) |\psi_1\rangle \quad (4.16)$$

Uma outra forma de expressar o operador descrito na equação (4.16) utilizando operadores quânticos elementares e o operador U é descrita na equação (4.17).

$$\exp\left(i \frac{\pi}{2n} \mathcal{H}\right) |\psi_1\rangle = \prod_{i=1}^n C U^{-2}_{cm_i} \prod_{j=1}^n U_{m_j} |\psi_1\rangle \quad (4.17)$$

Esta operação mede o número de zeros no registro \mathbf{m} , com um sinal positivo se c está no estado $|0\rangle$ e um sinal negativo se c está no estado $|1\rangle$. Aplicado ao estado $|\psi_1\rangle$ este operador irá determinar o número de qubits diferentes entre a entrada e os padrões armazenados na memória. Esta quantidade é denominada distância de Hamming.

O estado $|\psi_2\rangle$ é descrito na equação (4.18), onde $d_H(i, p_k)$ denota a distância de Hamming entre a entrada i e o padrão armazenado p^k .

$$|\psi_2\rangle = \frac{1}{\sqrt{2^p}} \sum_{k=1}^p \exp \left[i \frac{\pi}{2n} d_H(i, p^k) \right] |i_1, \dots, i_n; d_1^k, \dots, d_n^k; 0\rangle + \frac{1}{\sqrt{2^p}} \sum_{k=1}^p \exp \left[-i \frac{\pi}{2n} d_H(i, p^k) \right] |i_1, \dots, i_n; d_1^k, \dots, d_n^k; 1\rangle \quad (4.18)$$

O último passo determinístico do algoritmo é realizar uma operação quântica inversa a equação (4.14), para restaurar o registro de memória ao seu valor inicial e então aplicar o operador hadamard ao qubit de controle c .

$$|\psi_3\rangle = H_c \prod_k = n^1 XOR_{i_k m_k} NOT_{m_k} |\psi_2\rangle \quad (4.19)$$

Neste momento o estado $|\psi_3\rangle$ estará como descrito na equação (4.20), note que se a distância de Hamming tender a zero $d_H(i, p_k) \rightarrow 0$, então $\cos \left[\frac{\pi}{2n} d_H(i, p^k) \right] \rightarrow 1$ e $\sin \left[\frac{\pi}{2n} d_H(i, p^k) \right] \rightarrow 0$; e se $d_H(i, p_k) \rightarrow n$, então $\cos \left[\frac{\pi}{2n} d_H(i, p^k) \right] \rightarrow 0$ e $|\sin \left[\frac{\pi}{2n} d_H(i, p^k) \right]| \rightarrow 1$.

$$|\psi_3\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^p \cos \left[\frac{\pi}{2n} d_H(i, p^k) \right] |i_1, \dots, i_n; d_1^k, \dots, d_n^k; 0\rangle + \frac{1}{\sqrt{p}} \sum_{k=1}^p \sin \left[\frac{\pi}{2n} d_H(i, p^k) \right] |i_1, \dots, i_n; d_1^k, \dots, d_n^k; 1\rangle \quad (4.20)$$

Se um padrão de entrada for muito diferente de todos os padrões armazenados, então $d_H(i, p_k) \rightarrow n$ logo a medição do qubit de controle $|c\rangle$ irá retornar $|1\rangle$ com alta probabilidade, significando que o padrão não foi reconhecido, caso contrário, se um padrão próximo a todos os padrões armazenados, então $d_H(i, p_k) \rightarrow 0$ o que fará com que uma medição do qubit de controle $|c\rangle$ irá retornar $|0\rangle$ com alta probabilidade, significando que o padrão foi reconhecido. Caso o padrão seja reconhecido é realizada uma medição no registro de memória \mathbf{m} para recuperar o padrão.

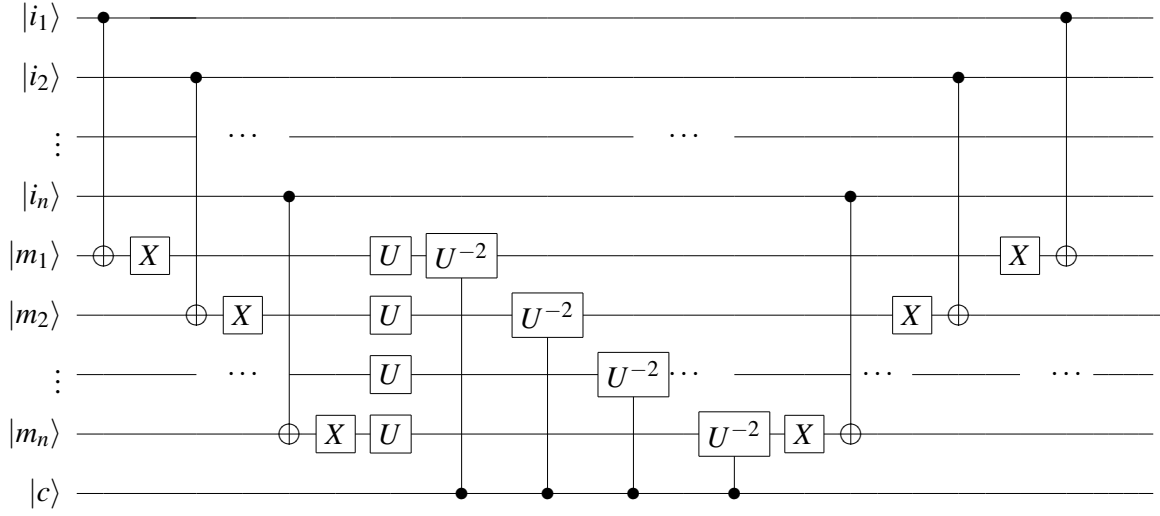


Figura 4.2 Circuito recuperação memória probabilística quântica

Para uma melhor compreensão do algoritmo de recuperação da memória QPM será ilustrado o seu funcionamento para $|m\rangle = \frac{1}{\sqrt{2}}(|00000\rangle + |11111\rangle)$, $|i\rangle = |10000\rangle$. Inicialmente é preparado o estado $|\psi_0\rangle = \frac{1}{\sqrt{2}}|i\rangle(|00000\rangle + |11111\rangle)|c\rangle$, onde $|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Em seguida o operador descrito na equação (4.14) é aplicado ao estado inicial obtendo-se $|\psi_1\rangle = \frac{1}{\sqrt{2}}|i\rangle(|01111\rangle + |10000\rangle)|c\rangle$. Em seguida é aplicado o operador descrito na equação (4.16), obtendo-se o estado descrito na equação (4.21).

$$\begin{aligned}
 |\psi_2\rangle = & \frac{1}{2} \left[\exp\left(\frac{i\pi}{2n}\right) |01111\rangle |0\rangle + \exp\left(\frac{i\pi}{2n} \cdot 4\right) |10000\rangle |0\rangle \right] + \\
 & \frac{1}{2} \left[\exp\left(\frac{-i\pi}{2n}\right) |01111\rangle |0\rangle + \exp\left(\frac{-i\pi}{2n} \cdot 4\right) |10000\rangle |0\rangle \right]
 \end{aligned} \tag{4.21}$$

A equação (4.19) descreve as últimas operações determinísticas do algoritmo de recuperação da informação. Após a sua aplicação obtém-se o estado descrito na equação (4.22).

$$\begin{aligned}
|\psi_3\rangle = & \frac{1}{2\sqrt{2}} \left[|i\rangle \left(\exp\left(\frac{i\pi}{2n}\right) |00000\rangle |0\rangle + \exp\left(\frac{i\pi}{2n} \cdot 4\right) |11111\rangle \right) |0\rangle + \right. \\
& \left. |i\rangle \left(\exp\left(\frac{i\pi}{2n}\right) |00000\rangle |0\rangle + \exp\left(\frac{i\pi}{2n} \cdot 4\right) |11111\rangle \right) |1\rangle \right] + \\
& \frac{1}{2\sqrt{2}} \left[|i\rangle \left(\exp\left(\frac{-i\pi}{2n}\right) |00000\rangle |1\rangle + \exp\left(\frac{-i\pi}{2n} \cdot 4\right) |11111\rangle \right) |1\rangle - \right. \\
& \left. |i\rangle \left(\exp\left(\frac{-i\pi}{2n}\right) |00000\rangle |0\rangle + \exp\left(\frac{-i\pi}{2n} \cdot 4\right) |11111\rangle \right) |1\rangle \right]
\end{aligned} \tag{4.22}$$

e como $\exp(ix) = \cos(x) + i\sin(x)$ e $n = 2$ pode-se verificar facilmente que $|\psi_3\rangle$ pode ser reescrito como na equação (4.23).

$$\begin{aligned}
|\psi_3\rangle = & \frac{1}{\sqrt{2}} \left[\cos\left(\frac{\pi}{4}\right) |i\rangle |00000\rangle |0\rangle + \cos(\pi) |i\rangle |11111\rangle |0\rangle + \right. \\
& \left. i \sin\left(\frac{\pi}{4}\right) |i\rangle |00000\rangle |1\rangle + i \sin(\pi) |i\rangle |11111\rangle |1\rangle \right]
\end{aligned} \tag{4.23}$$

Em seguida é realizada uma medição do registro **c** que resultará em $|0\rangle$ com probabilidade $\frac{1}{2} \cdot [\cos^2(\frac{\pi}{4}) + \cos^2(\pi)] = 0,75$ e resultará em $|1\rangle$ com probabilidade $\frac{1}{2} \cdot [\sin^2(\frac{\pi}{4}) + \sin^2(\pi)] = 0,25$.

Se o resultado obtido for igual a $|1\rangle$ o padrão não foi reconhecido e o processo de recuperação é reiniciado repetidamente no máximo por η vezes. Se em alguma destas repetições o resultado obtido for $|1\rangle$ o padrão foi reconhecido e é realizada uma medição no estado resultante $|\psi_4\rangle$ para recuperar a informação da memória.

É importante salientar que a medição no final do algoritmo faz com que a memória colapse para um dos valores armazenados e após esta medição a memória $|m\rangle$ não poderá ser reutilizada em um novo processo de recuperação.

Na próxima seção será descrito um modelo de rede neural quântica sem peso, utilizando a memória QPM, mas em seu funcionamento a medição no registro de memória **m** não é realizada.

4.4 Sumário do capítulo

Nas três primeiras seções deste capítulo o modelo de uma memória quântica probabilística, onde o tamanho da memória para armazenar todas as cadeias binárias com n qubits também é de n qubits. Logo o custo de armazenamento é de ordem linear, o que representa uma grande vantagem em comparação as memórias RAM que necessitam de 2^n bits para armazenar todas as cadeias binárias com n bits. Esta vantagem da memória QPM em relação a rede RAM é explorada para a definição de um modelo de rede neural sem peso onde o tamanho da memória não cresce exponencialmente com o número de entradas.

A criação de um estado quântico $|m\rangle$ onde todos os padrões do conjunto de treinamento estão em superposição pode permitir a criação de algoritmos de treinamento para redes neurais onde todos os padrões são apresentados a rede simultaneamente, implicando em uma drástica redução no tempo de execução dos algoritmos de treinamento. Esta redução do tempo necessário para o treinamento dos modelos neurais traria uma vantagem em diversas áreas que possuem como ponto negativo o custo computacional no treinamento de redes neurais, como por exemplo, o treinamento de redes neurais utilizando sistemas inteligentes híbridos.

Em particular um algoritmo de treinamento para a rede qPLN [39], onde as medições intermediárias são removidas, com uma única execução poderia ser criado utilizando a memória QPM para preprocessar os padrões do conjunto de treinamento em uma única entrada $|m\rangle$ a ser apresentada a rede e a utilização do algoritmo de recuperação da memória QPM para obter os valores desejados para os parâmetros da rede.

CAPÍTULO 5

Redes Neurais Artificiais

Neste capítulo são apresentados os modelos de redes neurais sem peso utilizados nesta dissertação. Na seção 5.1 são descritas as principais vantagens e desvantagens das RNSP. A seção 5.2 descreve o nodo RAM, que é a base dos modelos de RNSP utilizados neste capítulo e na seção 5.3 são descritas as redes RAM. As seções 5.4 e 5.5 descrevem o nodo lógico probabilístico (PLN) e redes PLN e a seção 5.6 descreve uma generalização do PLN. Nas seções 5.7 e 5.8 é descrito um modelo de RNSP onde os padrões do conjunto de treinamento são apresentados uma única vez durante a fase de treinamento. A seção 5.9 apresenta um método de transformação de números reais em binários. A seção 5.10 é um sumário do capítulo.

5.1 Introdução

Na maior parte da literatura, os modelos de redes neurais artificiais utilizados são baseados no nodo de McCulloch e Pitts [31], que possuem pesos associados às suas conexões. No entanto, neste capítulo serão descritos os modelos de redes neurais sem peso baseados em nodos RAM [5]. Os nodos de uma RNSP não possuem pesos ajustáveis associados às suas conexões, suas entradas e saídas são binárias e as funções dos nodos são armazenadas em tabelas verdades (posições de memória) que podem ser armazenadas em memórias de acesso aleatório (RAM). As RNSP também são conhecidas na literatura como redes neurais *RAM-based*, *n-tuple based* e redes neurais lógicas.

O treinamento de redes neurais sem peso consiste em modificar o conteúdo da memória dos nodos, este processo é mais rápido e flexível do que o utilizado nos modelos baseados no nodo de McCulloch e Pitts onde o algoritmo de treinamento utiliza técnicas de otimização como o gradiente descendente [8], que requerem o cálculo de derivadas. Além de possuir algoritmos de treinamento com rápida convergência, as RNSP podem ser facilmente implementadas em hardware [3] e não possuem limitações para a classificação de padrões que não são linearmente

separáveis.

Um dos maiores problemas destes modelos é a baixa taxa de generalização, um nodo RAM isolado não possui capacidade de generalizar, esta capacidade é obtida com a combinação de nodos [27]. Outro problema dos modelos neurais sem peso é que ao aumentar o número de entradas de um nodo do tipo RAM o tamanho de sua memória é incrementado exponencialmente.

Nas próximas seções serão descritos em detalhes o nodo RAM, seus variantes e redes que o possuem como unidade de processamento.

5.2 Nodo RAM

Com a finalidade de reconhecer padrões foi desenvolvido por Aleksander [4] o nodo RAM (Figura 5.2), que pode ser diretamente implementado em memória de acesso randômico disponível comercialmente. Um nodo RAM com n -entradas possui 2^n posições de memória endereçáveis por um vetor de n -bits $\mathbf{k} = (k_1, k_2, \dots, k_n)$. Um sinal binário $\mathbf{e} = (e_1, e_2, \dots, e_n)$ nas linhas de entrada da rede acessará apenas uma destas posições, onde $\mathbf{k} = \mathbf{e}$. O bit armazenado nesta posição de memória $C[\mathbf{k}]$ será a saída da rede $s = C[\mathbf{k}]$. Na definição 16 um nodo RAM é descrito formalmente.

Definição 16. [11] *Um nodo RAM é um neurônio artificial com as seguintes definições*

- $E = \{0, 1\}^n$ é o conjunto de entradas possíveis do nodo, onde n é a quantidade de conexões, também denominada fan-in;
- $K = \{0, 1\}^n$ é o conjunto de endereços do nodo. Para cada $\mathbf{k} \in K$, existe uma posição de memória $C[\mathbf{k}]$ que armazena a informação aprendida na forma de um bit. Um sinal binário $\mathbf{e} \in E$ no terminal de entrada irá acessar apenas a posição de memória $\mathbf{k} = \mathbf{e}$. O conteúdo $C[\mathbf{k}]$ acessado é chamado de conteúdo ativado.
- $s \in S$ é a saída do nodo, onde $S = \{0, 1\}$;
- $r \in S$ é o terminal de treinamento com a resposta desejada;
- $m \in \{0, 1\}$ é o terminal de modo de operação, indicando se o nodo está na fase de aprendizado ou uso.

- $f : E \rightarrow S$ é a função de transferência que computa s a partir do bit armazenado no endereço, determinado pelo terminal de entrada, sendo $s = f(C[k = e])$.

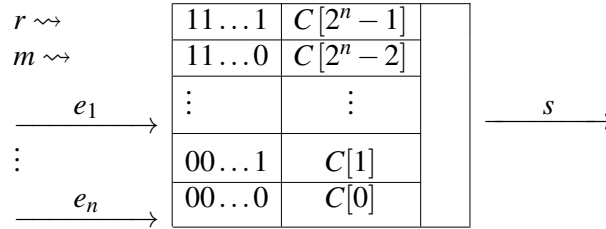


Figura 5.1 Nodo RAM

Um nodo RAM tem a capacidade de computar qualquer função lógica, enquanto nodos com peso podem computar apenas funções linearmente separáveis. Treinar um nodo RAM consiste apenas em alterar valores em sua memória, o que torna o seu treinamento muito rápido. Por ter entradas e saídas binárias e uma forma simples de funcionamento pode ser diretamente implementado em hardware.

Mesmo possuindo entradas e saídas binárias nodos RAM podem ser utilizados em problemas que possuem entradas contínuas através de técnicas de pré-processamento de dados contínuos em binários. Os nodos normalmente são parcialmente conectados, pois a memória do nodo cresce exponencialmente com o número de entradas n . A decisão sobre o valor de n influencia diretamente na capacidade de generalização de redes formadas por nodos RAM. Enquanto valores baixos levam a uma rápida saturação dos nodos, valores altos tendem a fazer com que os nodos se especializem nos padrões do conjunto de treinamento.

5.3 Rede RAM

A arquitetura de uma rede RAM típica é a de um grafo acíclico, com apenas uma camada *feedforward*. No algoritmo 3 todos os neurônios da rede têm suas posições de memória inicializadas com o valor 0, durante a fase de treinamento os conteúdos acessados, ao serem apresentados os exemplos de treinamento, têm seus valores alterados para 1. Um novo padrão apresentado a rede durante a fase de uso é classificado na mesma classe do conjunto de treinamento se todos os nodos tiverem 1 como saída.

Uma rede RAM pode ser utilizada para reconhecer apenas uma classe, quando mais classes são necessárias várias redes são combinadas, onde cada uma delas é responsável por discriminar uma classe. A resposta da rede é dada pela rede que possuir a maior saída. Cada rede do sistema é denominada discriminador e o conjunto de discriminadores é denominado multidiscriminador.

Um discriminador é composto por k nodos RAM com N entradas. Cada um destes k RAM's é endereçado por uma N -tupla (Figura 5.2), que é selecionada randomicamente de uma lista de valores binários, formando a entrada da rede. A saída do discriminador é a soma da saída dos nodos RAM's, produzindo a resposta r do discriminador.

Para treinar um discriminador pode ser utilizado um algoritmo que requer uma única apresentação do conjunto de treinamento (*one-shot*). O algoritmo 3 descreve um algoritmo de treinamento para um discriminador.

Algoritmo 3: Treinamento discriminador

```

1 Inicialize a memória dos nodos com o valor 0
2 para cada padrão  $p$  do conjunto de treinamento faça
3   | Apresente  $p$  ao discriminador
4   | Escreva o valor 1 nas posições de memórias ativadas por  $p$ 
5 fim
```

Um multidiscriminador é construído agrupando-se um conjunto de discriminadores, sendo cada um responsável por reconhecer uma classe de padrões diferentes. Para treinar um multidiscriminador é realizado o treinamento de cada um dos discriminadores que o compõe de forma independente.

5.4 Nodo Lógico Probabilístico (PLN)

Um aspecto importante dos nodos RAM é que existe uma ambiguidade nas posições de memória que armazenam o valor 0. Essa resposta pode tanto significar que tal vetor é um contra exemplo da classe ou que tal característica do padrão não foi treinada. Para eliminar esta ambiguidade foi desenvolvido em 1987 [20] o Nodo Lógico Probabilístico (PLN). A diferença entre um nodo PLN e um nodo RAM é que um número de 2 bits é armazenado em cada posição de memória do nodo e a saída do nodo é dada por uma função probabilística. O conteúdo das posições pode ser 1, 0 ou u , onde u representa um estado indefinido e é armazenado em todas

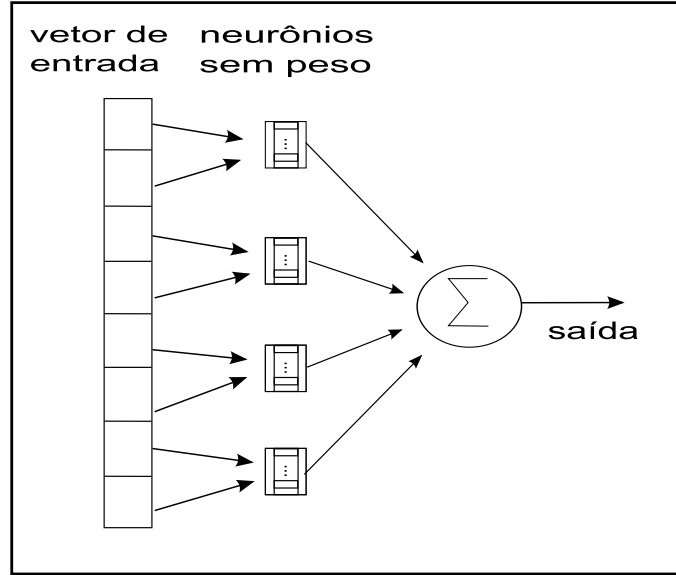


Figura 5.2 Discriminador

as posições de memória antes do treinamento do nodo PLN.

Quando uma posição de memória com o estado indefinido é acessado o nodo PLN tem 1 ou 0 como saída com igual probabilidade, a saída s de um nodo PLN é probabilística, podendo ser definida pela equação 5.1.

$$s = \begin{cases} 0, & \text{se } C[\mathbf{e}] = 0 \\ 1, & \text{se } C[\mathbf{e}] = 1 \\ \text{random}(0,1), & \text{se } C[\mathbf{e}] = u \end{cases} \quad (5.1)$$

onde $\text{random}(0,1)$ representa 0 ou 1 com igual probabilidade.

O treinamento de um nodo PLN consiste em alterar u 's para 0's ou 1's, onde a posição de memória $C[\mathbf{e}]$ recebe a saída s atual do nodo. Após a fase de treinamento se um nodo possuir posições com o valor indefinido u , ele poderá responder de forma aleatória para determinado padrão. Esta aleatoriedade na fase de uso de um nodo é indesejável na tarefa de reconhecimento de padrões, pois pode fazer que um mesmo padrão seja classificado de forma diferente ao ser apresentado a rede repetidamente. Então pode-se dizer que um nodo PLN *converge* quando não possui u 's em suas posições de memória ou se possuir que estes nunca sejam acessados.

5.5 Rede PLN

Enquanto redes RAM possuem arquitetura com apenas uma camada, na maioria dos trabalhos com redes PLN é utilizada uma arquitetura com múltiplas camadas, onde os nodos são dispostos em uma estrutura em forma de pirâmide (Figura 5.3), onde cada nodo é conectado a outros da camada imediatamente anterior, e a saída de cada nodo é conectada somente a um nodo. Se forem necessárias múltiplas saídas, podem ser utilizadas diversas pirâmides para compor a rede. As redes PLN são parcialmente conectadas, onde cada nodo possui baixa conectividade, para evitar que os nodos possuam uma memória com tamanho excessivo, no entanto um número muito baixo de entradas dos nodos da rede PLN pode fazer que a rede sature e não consiga aprender novos padrões antes que todo o conjunto de treinamento seja apresentado.

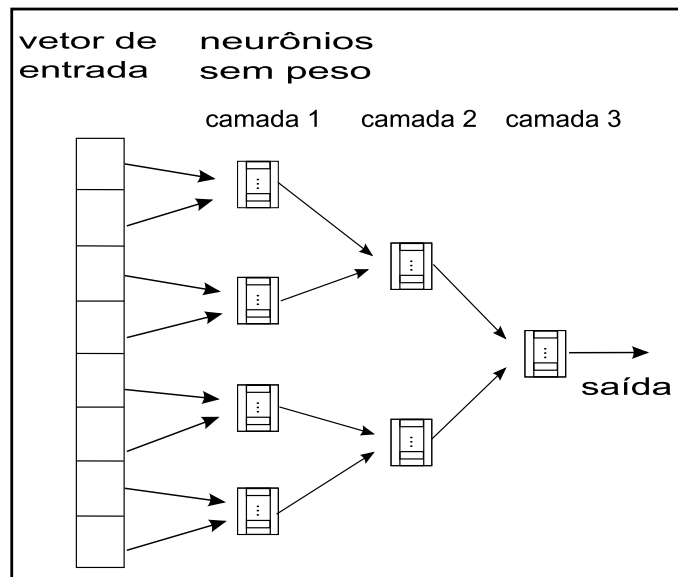


Figura 5.3 Rede PLN piramidal

O treinamento de redes PLN requer diversas apresentações do conjunto de treinamento, logo seu algoritmo de aprendizagem possui tempo computacional superior ao algoritmo de aprendizagem de redes RAM. No entanto a estrutura piramidal utilizada nas redes PLN aumentam a capacidade de generalização da rede.

O algoritmo 4 descreve o treinamento de redes PLN. O critério de parada no passo 2 pode ser o número de iterações, que deve ser escolhido de acordo com variáveis como conectividade dos nodos, número de nodos por camada, arquitetura da rede, entre outras e normalmente é definido através de um processo empírico; A estratégia de apresentação de padrões, passo 3, é

Algoritmo 4: Treinamento rede PLN

```

1  Inicialize todas as posições de memória com o valor  $u$ ;
2  enquanto Critério de parada não for alcançado faça
3      Apresente um padrão  $p$  do conjunto de treinamento à rede;
4      para  $t=1$  até  $\eta$  faça
5          Permita que a rede produza a saída  $s$  para o padrão  $p$ 
6          se  $s$  for igual a resposta desejada para o padrão  $p$  então
7               $aprender \leftarrow verdadeiro$ 
8              break
9          fim
10     fim
11     se  $aprender$  então
12         Cada nodo da rede aprende sua saída atual
13     senão
14         Escreva  $u$  nas posições de memória acessadas
15     fim
16 fim

```

outro fator que influencia no desempenho da rede, os padrões podem ser permutados após cada época e então apresentados sequencialmente ou podem ser escolhido aleatoriamente a cada apresentação; o loop com início no passo 4 checa se a rede está em um estado de indecisão, onde a rede tenta produzir a resposta desejada para o padrão selecionado η vezes, se esta for obtida com sucesso a rede aprende a saída desejada no passo 12; senão a rede entra em conflito de aprendizagem como descrito no passo 14.

5.6 Nodo Lógico Probabilístico Multivalorado (MPLN)

A diferença entre o nodo MPLN e o nodo PLN é que o primeiro permite um maior, mas ainda discreto, intervalo de probabilidades a ser armazenado em cada posição de memória. A saída de um nodo MPLN é probabilística, sendo igual a 1 com probabilidade $C[e]$ e 0 com probabilidade $1 - C[e]$.

O treinamento de um nodo MPLN é realizado de forma mais gradual que de um nodo PLN e é realizado através de alterações no conteúdo da memória do nodo através de uma recompensa

ou punição como descrito na equação 5.2.

$$C[\mathbf{e}] = \begin{cases} C[\mathbf{e}] - \eta g(d) & \text{punição} \\ C[\mathbf{e}] + \eta g(d) & \text{recompensa} \end{cases} \quad (5.2)$$

onde \mathbf{e} é a entrada do nodo, $C[\mathbf{e}]$ é a posição de memória acessada, d é a resposta desejada e $g : \{0, 1\} \rightarrow \{-1, 1\}$ é uma função descrita pela equação 5.3.

$$g(d) = \begin{cases} +1 & \text{se } d = 1 \\ -1 & \text{se } d = 0 \end{cases} \quad (5.3)$$

A vantagem da rede MPLN em relação a rede PLN é que após diversos reforços em uma direção, é difícil apagar o conhecimento obtido. De fato, será necessário o mesmo número de experiências negativas para retornar ao valor u [5], desta forma uma rede composta por nodos MPLN é menos sensível a ruídos que redes formadas por nodos PLN.

O treinamento da rede MPLN é descrito no algoritmo 5 e segue a mesma idéia do algoritmo de treinamento da rede PLN, onde apenas a forma de atualização do conteúdo das posições de memória do nodo é alterado. O valor da saída desejada d dos nodos das camadas ocultas é a saída s atual do nodo.

Algoritmo 5: Treinamento rede MPLN piramidal

```

1 Inicialize todas as posições de memória com o valor 0.5;
2 enquanto Critério de parada não for alcançado faça
3   Apresente um padrão  $p$  do conjunto de treinamento à rede;
4   para  $t=1$  até  $\eta$  faça
5     Permita que a rede produza a saída  $s$  para o padrão  $p$ 
6     se  $s$  for igual a resposta desejada para o padrão  $p$  então
7        $aprender \leftarrow verdadeiro$ 
8       break
9     fim
10  fim
11  se  $aprender$  então
12    Reforçar todos os nodos da rede
13  senão
14    Punir todos os nodos da rede
15  fim
16   $aprender \leftarrow falso$ 
17 fim

```

Uma rede lógica converge quando todos os conteúdos de memória, em todos os nodos, têm probabilidade de ter 1 como saída igual a 1 ou 0, ou então quando o conteúdo nunca for endereçado. Após a convergência, uma rede MPLN pode ser substituída por uma rede PLN equivalente sem alteração da performance.

5.7 Nodo GSN

O goal seeking neuron (GSN), foi proposto por Filho [10] com a intenção de superar as limitações do nodo PLN, uma rede formada por nodos GSN possui resposta determinística e faz um uso mais eficiente do espaço de memória limitado do nodo RAM [7]. A principal característica do nodo GSN é que seu treinamento é realizado de forma que cada padrão do conjunto de treinamento é apresentado apenas uma vez a rede (one-shot learning), o que leva a um tempo de treinamento muito baixo.

O nodo GSN é capaz de armazenar, receber e gerar os valores 0, 1 e u . Ao receber um valor u o nodo terá mais que uma posição de memória acessada, por exemplo se um nodo com duas entradas receber $0u$, então serão acessadas as posições 01 e 00, note que se a entrada deste nodo for uu todas as posições de memória serão acessadas. Durante o treinamento a saída do nodo será 0 quando todas as posições de memória acessadas contiverem o valor 0, da mesma forma a saída do nodo será 1 quando todas as posições de memória acessadas contiverem o valor 1, caso contrário a saída será um valor indefinido u .

Os nodos deste tipo de rede neural possuem três estados de operação: O *estado de validação* onde o nodo verifica se um novo padrão apresentado pode ser aprendido; o *estado de aprendizagem*, onde ocorre a alteração das posições de memória do nodo; e o *estado de uso* em que o nodo é utilizado para produzir uma saída para novos padrões.

5.7.1 Estado de validação

O estado de validação é utilizado durante o treinamento, o objetivo do nodo neste momento é definir que valores podem ser aprendidos para uma dada entrada e sem interferir no conhecimento previamente adquirido pela rede [9]. Durante este estado o valor da saída s do nodo será 0 ou 1 apenas se todas as posições acessadas por e foram iguais a 0 ou 1, caso contrário s será igual a u . Se a saída do nodo for um valor indefinido u o nodo poderá aprender qualquer valor

desejado, caso contrário se a saída for um valor definido (0 ou 1) o nodo só poderá aprender o valor obtido em sua saída.

5.7.2 Estado de Aprendizagem

No estado de aprendizagem o nodo inicialmente procura por uma das posições de memória acessadas na fase de validação que possua o mesmo valor que a resposta desejada. Caso este valor não seja encontrado, uma posição com valor indefinido u tem seu conteúdo alterado para a resposta desejada d . Caso exista mais que uma opção contendo a resposta desejada ou com valores indefinidos uma escolha aleatória é realizada para decidir que posição de memória será utilizada. A posição de memória escolhida irá definir quais as respostas desejadas dos nodos que possuem conexões com o nodo atual na camada anterior.

5.7.3 Estado de uso

No estado de uso o nodo tem como objetivo produzir o valor binário com maior número de ocorrências no conteúdo acessado, o nodo terá como saída s um valor indefinido u apenas se o número de 0's e 1's acessados forem iguais ou se houver apenas u 's nas posições de memória acessadas.

5.8 Rede GSN

Assim como nas redes PLN e MPLN, na maioria dos trabalhos, as redes GSN possuem arquitetura de múltiplas camadas onde os nodos normalmente ficam dispostos em forma de pirâmide. No entanto a arquitetura de uma rede GSN pode ser definida durante o treinamento da rede [30]. Enquanto um nodo GSN possui três estados de funcionamento, uma rede GSN possui duas fases de processamento: aprendizagem e de uso.

Cada iteração da fase de aprendizagem possui dois estágios: no primeiro estágio os nodos encontram-se no estado de validação, um padrão é apresentado a rede e esta trabalha para frente para produzir a saída correspondente à entrada p . Como os nodos estão em fase de validação a rede tende a produzir valores indefinidos u ; a rede entra no segundo estágio da fase de aprendizagem se a saída da rede s for um valor indefinido u ou se s for igual a resposta desejada $d[p]$,

neste estágio os nodos estão na fase de aprendizagem e a rede processa a informação de trás para frente. O algoritmo 6 descreve o aprendizado em redes GSN.

No processo de uso os nodos estão na fase de uso e o processamento da rede ocorre sempre da primeira camada para a última. Quando um padrão é apresentado a rede esta tende a responder com 0 ou 1, mas a saída de um valor indefinido pode ocorrer, neste caso a rede rejeita o padrão apresentado e não o classifica.

Algoritmo 6: Aprendizado rede GSN

```

1 Inicialize todo conteudo das posições de memória dos nodos com valores indefinidos  $u$ 
2 para cada padrão  $p$  do conjunto de treinamento faça
3   estado da rede  $\leftarrow$  validação;
4   Apresente  $p$  e permita que a rede produza a saída  $s$  para o padrão apresentado;
5   se  $s = u$  ou  $s = d[p]$  então
6     estado da rede  $\leftarrow$  aprendizagem;
7     Permita que a rede aprenda a resposta desejada  $d[p]$ .
8   fim
9 fim

```

5.9 Pré-processamento

Os modelos neurais estudados neste capítulo só podem ter como entradas valores Booleanos, o que limita a aplicabilidade das RNSP. No entanto diversas formas para a transformação de números reais em Booleanos são conhecidas. Nesta seção serão descritas a técnica de quantização denominada CMAC e a técnica para binarização conhecida por escala de cinza, que produzem números binários com a característica de preservar as distancias entre os padrões, i.e, proximidade numérica é convertida em proximidade na distância de Hamming, esta característica é importante, pois proximidade na distância de Hamming tem papel fundamental na generalização nas redes neurais sem peso [45].

Para codificar um número real x em binário utilizando CMAC e escala de cinza, primeiro defini-se um número $K \in \mathbb{Z}$, então são calculados K números onde o j -ésimo é dado por $m_j = (x + j - 1)/K$ arredondado para baixo e expresso em escala de cinza. A escala de cinza de um inteiro i pode ser obtida realizando um ou exclusivo bit a bit entre a representação binária de i e de $i/2$. Isto irá prover uma representação binária de tal forma que se dois números x e y tiverem sua diferença $x - y \leq K$ então a distância de Hamming de seus códigos é de $|x - y|$, e

se a distância de Hamming for maior ou igual a K , a distância de Hamming é pelo menos K .

5.10 Sumário do capítulo

Este capítulo teve como objetivo apresentar os modelos de redes neurais sem peso utilizados nesta dissertação. No decorrer do capítulo foram descritos os nodos e as redes sem peso RAM, PLN, MPLN e GSN. As principais vantagens e desvantagens de cada modelo foram apresentadas, para dar uma visão geral sobre os modelos de redes neurais sem peso, assim como para os seus algoritmos de treinamento.

Na seção 5.9 foram descritas técnicas que permitem que dados reais sejam transformados em dados binários, isto permite que as redes booleanas sejam utilizadas em uma gama maior de problemas.

Redes Neurais Quânticas

Neste capítulo serão descritos alguns dos modelos de redes neurais quânticas e seus algoritmos de treinamento. Será verificado que a maioria dos modelos apresenta um problema comum no algoritmo de treinamento, a utilização de operadores não lineares ou não unitários. Esta quebra do postulado 2 da mecânica quântica irá impossibilitar a execução destes algoritmos em um hipotético computador quântico. O capítulo está dividido em quatro subseções. A seção 6.1 introduz o conceito de computação quântica neural e propõe uma classificação para os algoritmos de aprendizado quânticos. Nas seções 6.2 e 6.3 são analisados, respectivamente, os algoritmos quânticos iterativos e baseados na superposição. A seção 6.4 é um sumário do capítulo.

6.1 Introdução

O conceito de computação quântica neural foi introduzido por Subhash Kak em 1995, criando um novo paradigma utilizando redes neurais e computação quântica e abrindo diversas direções na pesquisa em redes neurais [19]. Espera-se que as redes neurais quânticas sejam mais eficientes que as redes neurais clássicas, assim como é esperado entre a computação quântica e clássica [49].

No mesmo ano, Menneer e Narayanan propôs um modelo de rede neural inspirado na computação quântica, com redes de única camada, utilizando o ponto de vista de múltiplos universos [32]. Em 1998 Ventura e Martinez introduziram a memória associativa quântica, com uma capacidade exponencial no número de neurônios [58].

A não linearidade das funções de ativação utilizadas nos modelos de redes neurais favoreceu ao atraso do desenvolvimento das RNQ, mas em 2001 Altaisky propôs um sistema quântico onde as regras do aprendizado do perceptron podem ser utilizados [6]. Recentemente, diversos modelos de redes neurais quânticas têm sido propostos, mas permanece o desafio da imple-

mentação direta em circuitos quânticos, adaptação natural dos algoritmos de aprendizado e possibilidade física do aprendizado [39].

Por ser uma área de pesquisa recente, as redes neurais quânticas não possuem seus modelos bem definidos e os algoritmos de treinamento possuem problemas em aberto. Como será visto nas seções 6.2.1, 6.2.2, 6.2.3 existem diversas tentativas de quantização do perceptron, mas todas apresentam algum problema em relação aos postulados da mecânica quântica.

Enquanto o perceptron possui diversas modelagens quânticas, as redes neurais sem peso quânticas são inicialmente propostas em 2007 e 2008 nos trabalhos de Oliveira [38], [39]. Como será visto nas seções 7.1 e 7.2 são definidos versões quânticas do neurônio lógico probabilístico e do neurônio lógico probabilístico multi-valorado, que respeitam os princípios da computação quântica e os algoritmos de aprendizado clássicos podem ser adaptados para a versão quântica dos modelos.

Os algoritmos de aprendizado das redes neurais quânticas propostos até o momento podem ser classificados em iterativos e baseados na superposição. Nas próximas subseções será realizada uma análise dos algoritmos de treinamento das RNQ.

6.2 Algoritmos iterativos

Nesta seção serão descritos alguns dos modelos de redes neurais quânticas e seus algoritmos de treinamento propostos entre o ano de 2001 e 2010 e que possuem algoritmos de aprendizado iterativos. Será verificado que a maioria dos modelos possuem problemas quanto a implementação em circuitos quânticos, adaptação dos algoritmos de aprendizado clássicos e criação de algoritmos quânticos que respeitem os postulados da mecânica quântica.

6.2.1 Neurônio artificial quântico

Dan Ventura e Tony Martinez introduzem um modelo matemático de um neurônio artificial com propriedades quânticas [57] que tem a capacidade de classificar padrões não linearmente separáveis utilizando apenas funções de ativação lineares. Partindo de um perceptron e adicionando propriedades quânticas ao modelo. Neste modelo os pesos w são substituídos por funções de onda, i.e., qubits. Então o vetor de pesos é substituído por qubits em superposição, que ao interagir com o ambiente irão colapsar para um vetor de pesos clássico.

Apesar de mostrar a possibilidade de classificação de padrões e sugerir uma regra de aprendizado Ventura e Martinez concluem seu trabalho indicando que existe muito a ser feito, como o desenvolvimento de um algoritmo de aprendizado e análise teórica das capacidades do neurônio quântico.

Ventura e Martinez propuseram uma das primeiras modelagens utilizando propriedades quânticas do perceptron. A idéia de utilizar qubits como pesos do neurônio foi aparentemente abandonada e os demais modelos utilizam operadores como pesos e qubits como as entradas do neurônio.

6.2.2 Regra delta quântica

Um dos primeiros modelos quântico para o perceptron foi proposto em 2001 por Altaisky [6]. Os pesos do perceptron de Altaisky são operadores quânticos sobre um único qubit e sua regra de aprendizado utiliza funções de ativação lineares. Altaisky afirma que dificilmente serão construídas análogas quânticas das funções de ativação não lineares usualmente utilizadas em redes neurais.

O trabalho de Altaisky [6] é um dos primeiros a tratar sobre uma versão quântica do perceptron, utilizando operadores como os pesos, e de sua regra de aprendizado. Neste é afirmado que dificilmente serão construídas análogas quânticas das funções de ativação não lineares usualmente utilizadas em redes neurais. Mas Altaisky propõe uma versão quântica do perceptron com função de ativação linear e uma regra de aprendizado por correção do erro.

Um perceptron em um sistema quântico com entradas $|x_1\rangle, \dots, |x_n\rangle$ terá sua saída $|y\rangle$ definida pela equação 6.1,

$$|y\rangle = \hat{F} \sum_{j=1}^n \hat{w}_j |x_j\rangle, \quad (6.1)$$

onde \hat{w}_j são matrizes 2×2 , \hat{F} é um operador quântico.

A regra de aprendizado é mostrado considerando $\hat{F} = I$, onde a saída no tempo t é dada por:

$$|y(t)\rangle = \sum_{j=1}^n \hat{w}_j(t) |x_j\rangle, \quad (6.2)$$

A regra de aprendizado Com a seguinte regra de aprendizado por correção do erro

$$\hat{w}_j(t+1) = \hat{w}_j(t) + \eta(|d\rangle - |y(t)\rangle)\langle x_j| \quad (6.3)$$

onde $|d\rangle$ é a resposta desejada

Teorema 6. *A regra de aprendizado 6.3 altera as matrizes \hat{w}_j de forma a conduzir a saída $|y\rangle$ do perceptron com a função de ativação I a resposta desejada $|d\rangle$*

Demonstração. Considere o módulo quadrático da diferença entre a saída real e a desejada. Pela equação 6.1

$$|| |d\rangle - |y(t+1)\rangle ||^2 = || |d\rangle - \sum_{j=1}^n \hat{w}_j(t+1) |x_j\rangle ||^2$$

E pela equação 6.3

$$\begin{aligned} || |d\rangle - \sum_{j=1}^n \hat{w}_j(t+1) |x_j\rangle ||^2 &= || |d\rangle - \sum_{j=1}^n [\hat{w}_j(t) + \eta(|d\rangle - |y(t)\rangle)\langle x_j|] |x_j\rangle ||^2 = \\ || |d\rangle - \sum_{j=1}^n [\hat{w}_j(t) |x_j\rangle + \eta(|d\rangle - |y(t)\rangle)\langle x_j| |x_j\rangle] ||^2 &= || |d\rangle - \sum_{j=1}^n [\hat{w}_j(t) |x_j\rangle] + n\eta(|d\rangle - |y(t)\rangle) ||^2 = \\ || |d\rangle - |y(t)\rangle + n\eta(|d\rangle - |y(t)\rangle) ||^2 &= (1 - n\eta)^2 || |d\rangle - |y(t)\rangle ||^2 \end{aligned}$$

Logo

$$|| |d\rangle - |y(t+1)\rangle ||^2 = (1 - n\eta)^2 || |d\rangle - |y(t)\rangle ||^2$$

e considerando $0 < \eta < 1/n$

$$|| |d\rangle - |y(t+1)\rangle ||^2 < || |d\rangle - |y(t)\rangle ||^2$$

□

A regra expressa na equação 6.3 conduz o sistema a resposta desejada não preserva a unitariedade dos operadores \hat{w}_j durante o aprendizado [6], desrespeitando o postulado 2 da mecânica quântica.

Lema 2. *A regra de aprendizado descrita na equação 6.3 não preserva a unitariedade dos operadores \hat{w}_j .*

A prova será realizada por contra exemplo. Para isto será realizada uma iteração do algoritmo de aprendizado para um peso da rede, considere na equação 6.3 que $j = 1$, o peso $\hat{w}_1(t) = I$, que a resposta desejada seja $|d\rangle = |1\rangle$, a saída atual da rede é dada por $|0\rangle$ e que a entrada associada a $\hat{w}_1(t)$ seja $|x_j\rangle$ e $\eta = 0.5$. Tem-se então que

$$\begin{aligned}\hat{w}_j(t+1) &= I + 0,5(|1\rangle - |0\rangle)\langle 1| = \hat{w}_j(t+1) = I + 0,5(|1\rangle\langle 1| - |0\rangle\langle 1|) = \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & -0.5 \\ 0 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & -0.5 \\ 0 & 1.5 \end{pmatrix}\end{aligned}$$

Conclui-se então que $w_1(t+1)$ é não unitária \square .

Uma tentativa para manter a unitariedade dos operadores é tentar escolher uma taxa de aprendizado que mantenha a unitariedade, mas esta técnica não resolverá o problema. Isto pode ser visto voltando a demonstração do lema anterior. Se utilizarmos uma taxa de aprendizado arbitrária chegar-se-ia a

$$\hat{w}_1(t+1) = \begin{pmatrix} 1 & -\eta \\ 0 & 1+\eta \end{pmatrix} \leftarrow \hat{w}_1(t+1)\hat{w}_1(t+1)^\dagger = \begin{pmatrix} 1+\eta^2 & -\eta-\eta^2 \\ -\eta-\eta^2 & (1+\eta)^2 \end{pmatrix}$$

e segundo a condição de unitariedade

$$\begin{pmatrix} 1+\eta^2 & -\eta-\eta^2 \\ -\eta-\eta^2 & (1+\eta)^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

de onde se conclui que $\eta = 0$, ou seja, para este caso não existe taxa de aprendizado não nula que permita que a regra expressa na equação 6.3 seja realizada sem quebrar o postulado 2 da mecânica quântica. Isto evidencia que a regra proposta por Altaisky não pode ser utilizada em um computador quântico, considerando que os atuais postulados da mecânica quântica estejam corretos.

6.2.3 Rede neural quântica M-P

Outro modelo para um perceptron quântico é proposto por Rigui Zhou e Qiulin Ding [62]. Um perceptron quântico com n entradas, $|x_1\rangle, \dots, |x_n\rangle$, tem seus pesos representados por uma

única matriz W de dimensão $2^n \times 2^n$, i.e., um operador quântico sobre n qubits. A saída $|y\rangle$ será determinada da seguinte forma:

$$|y\rangle = W |x_1, \dots, x_n\rangle \quad (6.4)$$

O algoritmo de aprendizado da rede neural quântica M-P é apresentado abaixo:

1. Inicialize a matriz de pesos W^0 .
2. Dado um conjunto de pares de entrada-saída $(|\varphi\rangle, |O\rangle)$.
3. Calcule a saída $|\psi\rangle = W^t |\varphi\rangle$.
4. Atualize os pesos da rede $w_{ij}^{t+1} = w_{ij}^t + \eta(|O\rangle_i - |\psi\rangle_i) |\varphi\rangle_j$.
5. repita 3 e 4 até obter erros aceitáveis.

Existem alguns problemas no processo de atualização dos pesos w_{ij} no algoritmo de aprendizado da rede neural quântica M-P. No 6 passo do algoritmo a i -ésima amplitude dos qubits $|O\rangle$ e $|\varphi\rangle$ são acessadas. Além disso, a regra mantém o problema da não unitariedade.

Para um melhor entendimento do algoritmo e para verificar o problema com a unitariedade segue um exemplo, mostrado em [62], de uma iteração do algoritmo.

- Suponha $W^0 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$;
- O par de entrada-saída $\left(|\varphi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)\right)$
- A saída atual da rede é dada por $|\psi\rangle = W^0 |\varphi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- A nova matriz de pesos W^1 é calculada da seguinte forma:

$$w_{00}^1 = w_{00}^0 + \eta(|O\rangle_0 - |\psi\rangle_0) |\varphi\rangle_0 = -1 + \left(\frac{-1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) \frac{1}{\sqrt{2}} = -1$$

$$w_{01}^1 = w_{01}^0 + \eta(|O\rangle_0 - |\psi\rangle_0) |\varphi\rangle_1 = -1 + \left(\frac{-1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) \frac{1}{\sqrt{2}} = -1$$

$$w_{10}^1 = w_{10}^0 + \eta(|O\rangle_1 - |\psi\rangle_1) |\varphi\rangle_0 = -1 + \left(\frac{-1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) \frac{1}{\sqrt{2}} = -1$$

$$w_{11}^1 = w_{11}^0 + \eta(|O\rangle_1 - |\psi\rangle_1) |\varphi\rangle_1 = -1 + \left(\frac{-1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) \frac{1}{\sqrt{2}} = -1$$

Então a matriz de pesos W^1 é:

$$W^1 = \begin{pmatrix} -1 & -2 \\ 1 & 0 \end{pmatrix}$$

que evidentemente não é unitária. Ocorrendo uma quebra do postulado 2 na primeira iteração do algoritmo.

A estrutura da rede neural M-P permite que um neurônio possa separar padrões não linearmente separáveis como, por exemplo, a paridade de 2 bits. Para isto basta utilizar como matriz de pesos a porta não controlado

$$W = C_{not} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

6.3 Algoritmos baseados na superposição

Os modelos de redes neurais quânticas apresentados até o momento possuem algoritmo de treinamento iterativo. Além da utilização de operadores não unitários os modelos iterativos não possuem uma divisão clara entre os dados de entrada e saída, pois estes ocupam os mesmos registradores.

Uma forma de apresentação dos padrões de treinamento em superposição em problemas de aprendizagem de máquina foi proposta por Ventura em (Ventura e Martinez, 1999). Onde um algoritmo quântico para criar um estado superposto contendo os padrões de um conjunto do conjunto de treinamento é proposto. Este algoritmo é utilizado para inicialização de dados em diversos modelos de redes neurais.

Em [44] este algoritmo é utilizado para inicializar dados em um conjunto de treinamento que em seguida são apresentados a uma rede neural com parâmetros em superposição. Em seguida o algoritmo de busca de Grover é utilizado para treinar a rede neural. Este método é utilizado para realizar o treinamento de redes neurais baseadas no perceptron de múltiplas camadas [44], [41], redes neurais sem peso [51] entre outros modelos de redes neurais quânticas [40].

6.4 Sumário do capítulo

Neste capítulo foram apresentados alguns modelos de redes neurais quânticas. Os modelos de perceptron quânticos apresentam como principal vantagem a capacidade de classificar corretamente padrões não linearmente separáveis. A análise dos algoritmos quânticos propostos até o momento mostram que estes apresentam problemas em relação aos postulados da mecânica quântica, pois utilizam operadores não lineares ou não unitários.

Neurônios Lógicos Quânticos

Neste capítulo serão definidos os modelos de redes neurais sem peso quânticas propostos nesta dissertação. Nas seções 7.1, 7.2 e 7.3 são descritos os modelos de redes neurais sem peso quânticas baseados, respectivamente, nas redes PLN, MPLN e RAM. Os modelos propostos são denominados, neurônio lógico probabilístico quântico (qPLN), neurônio lógico probabilístico quântico multivalorado (qMPLN) e neurônio RAM quântico (qRAM). A definição dos modelos será realizada de forma a manter as propriedades das redes neurais sem peso e permitir a extensão do estudo destes modelos à computação quântica. Na seção 7.3.2 um algoritmo de treinamento para redes neurais quânticas baseado na superposição (SLA, do inglês Superposition based Learning Algorithm) será proposto e em seguida é apresentada uma alteração do algoritmo SLA que permite o treinamento da rede com apenas duas execuções da rede. A vantagem dos algoritmos SLA em relação aos algoritmos clássicos é a capacidade de receber todos os padrões do conjunto de treinamento simultaneamente, utilizando o princípio da superposição. Na seção 7.4 são realizadas considerações sobre a universalidade dos modelos propostos o que sugere a criação de um novo modelo. Finalmente a seção 7.5 é um sumário do capítulo. As principais contribuições deste trabalho são descritas nas seções 7.3 e 7.3.2.

7.1 qPLN

O primeiro passo para definir o neurônio qPLN é associar os valores 0, 1 e u , respectivamente, com os qubits $|0\rangle$, $|1\rangle$ e $H|0\rangle = |u\rangle$. Substituindo os valores clássicos pelos correspondentes quânticos na definição do neurônio PLN e o gerador de saída probabilístico pela medição do qubit correspondente, obtém-se o neurônio qPLN com a saída descrita na equação (7.1).

$$r = \begin{cases} 0, & \text{if } C[I] = |0\rangle \\ 1, & \text{if } C[I] = |1\rangle \\ \text{random}(0,1), & \text{if } C[I] = |u\rangle \end{cases} \quad (7.1)$$

Pelas equações (7.1) e (5.1) pode se verificar que existe uma bijeção entre as saídas de um PLN e do qPLN que associa i a $|i\rangle$, onde $i = 0, 1$ ou u . A figura 7.1 mostra um esquema de um neurônio qPLN, onde qubits são “armazenados” nas posições de memória e qubits são dados como entrada do neurônio. Para implementar este modelo precisamos resolver três problemas: (1) Quando for realizada uma medição em uma posição de memória que contenha um estado quântico ocorrerá perda de informação; (2) Como realizar endereçamento quando existir estados quânticos nas linhas de entrada do neurônio; (3) Como treinar uma rede composta por neurônios qPLN.

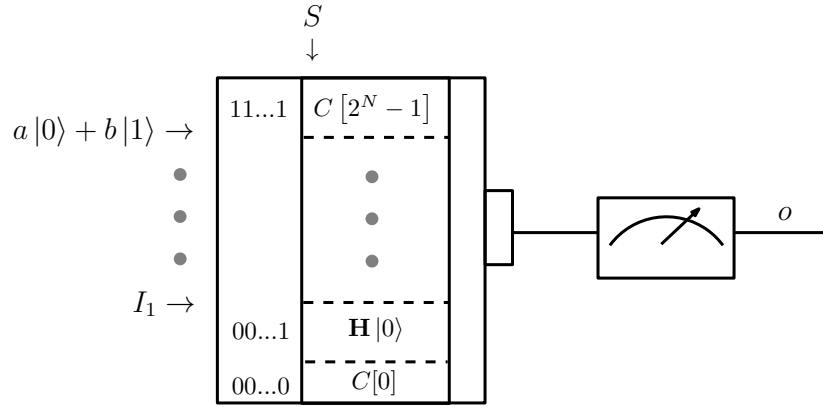
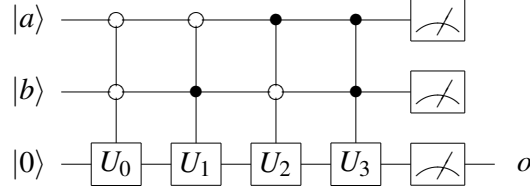


Figura 7.1 Esquema neurônio qPLN

Mostraremos, sem perda de generalidade, um neurônio com 2 entradas, figura 7.2, que solucionará os itens 1 e 2 citados acima. Onde ao invés de armazenar os estados quânticos necessários são armazenados operadores que ao agirem sobre o qubit $|0\rangle$ produzem a saída desejada, o endereçamento é realizado através de operadores quânticos controlados.

Utilizando o circuito da figura 7.2 podemos “armazenar” os qubits sem perda de informações após medição, pois a cada execução do neurônio o estado será recriado pelo operador U_i e o endereçamento é realizado através do uso de controles sobre os operadores U_i . O terceiro problema (o treinamento da rede) citado anteriormente permanece, pois o treinamento do neurônio qPLN seria uma troca de operadores, ou seja, uma alteração do circuito do neurônio. Portas quânticas programáveis determinísticas universais não são possíveis em geral [23], mas o neurônio qPLN requer apenas três tipos de matriz. Desta forma podemos definir um operador



onde

$$U_i = I, X \text{ or } H$$

Figura 7.2 Neurônio qPLN em circuito quântico

A , como na equação (7.2),

$$A = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & H & 0 \\ 0 & 0 & 0 & U \end{pmatrix} \quad \text{onde} \quad \begin{aligned} A|000\rangle &= |00\rangle I|0\rangle, \\ A|010\rangle &= |01\rangle X|0\rangle, \\ A|100\rangle &= |10\rangle H|0\rangle, \end{aligned} \quad (7.2)$$

que opera em 2 qubits adicionais (chamados seletores) e produz a saída de qualquer matriz U_i , apenas ajustando os valores dos seletores. Uma outra forma de visualizar a matriz A é utilizando a representação de produto externo $A = |00\rangle\langle 00| \otimes I + |01\rangle\langle 01| \otimes X + |10\rangle\langle 10| \otimes H + |11\rangle\langle 11| \otimes U$.

Substituindo na figura 7.2 cada matriz U_i por matrizes A com acréscimo de 2 registros de entrada para cada matriz obtém-se o neurônio qPLN descrito na figura 7.3. Cada operador A_i tem a mesma ação do operador A com seletores s_j^i e qubit alvo $|o\rangle$. Desta forma o treinamento do neurônio é obtido simplesmente pela alteração dos seletores.

Esta quantização do neurônio PLN permite uma adaptação, com pequenas modificações, do algoritmo de treinamento do PLN, o algoritmo 7 descreve o treinamento de uma rede piramidal formada por neurônios qPLN. Note que este algoritmo é uma simples adaptação do algoritmo clássico para as redes qPLN e não utiliza propriedades da mecânica quântica. Sua utilidade é apenas mostrar que uma rede qPLN pode simular uma rede PLN.

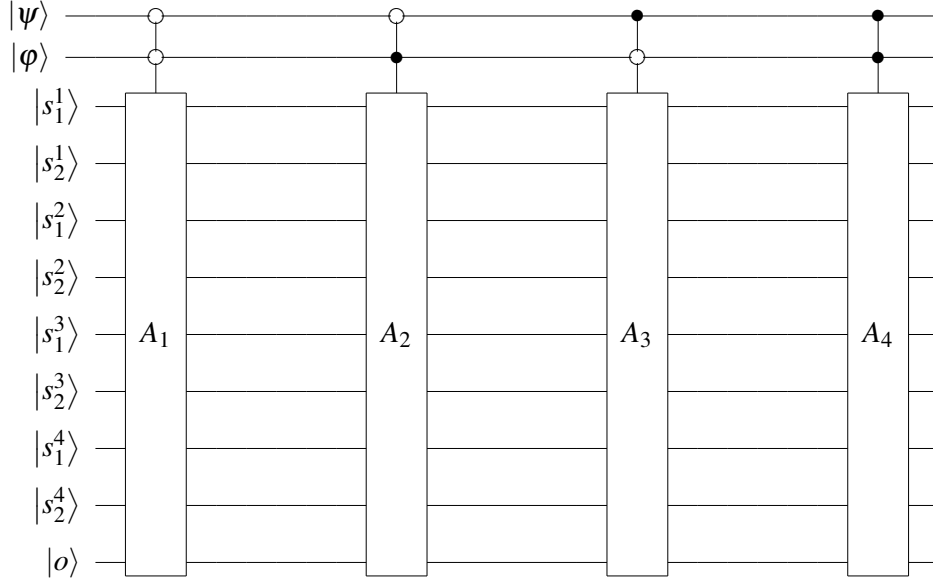


Figura 7.3 Neurônio qPLN com seletores

7.2 qMPLN

No caso clássico, a única diferença entre o neurônio PLN e o MPLN é que o último pode armazenar um número maior de probabilidades em cada posição de memória. Em um neurônio PLN pode-se armazenar apenas o conjunto $\{0, 1, u\}$ e no neurônio MPLN é possível armazenar um conjunto finito de probabilidades em geral maior que $\{0, 1, u\}$ [27]. O neurônio qMPLN será definido de forma que a mesma diferença poderá ser notada entre o qPLN e o qMPLN. Para isto considere a matriz U_p definida na equação (7.3).

$$U_p = \begin{pmatrix} \sqrt{1-p} & \sqrt{p} \\ \sqrt{p} & -\sqrt{1-p} \end{pmatrix} \quad (7.3)$$

Com esta notação pode-se verificar facilmente que a medição de $U_0|0\rangle = |0\rangle$, $U_1|0\rangle = |1\rangle$ e $U_{\frac{1}{2}}|0\rangle = |u\rangle$ são os valores correspondentes em um neurônio qPLN. Note que a medição de $U_p|0\rangle$ irá retornar 1 com probabilidade p , pois

$$U_p|0\rangle = \begin{pmatrix} \sqrt{1-p} & \sqrt{p} \\ \sqrt{p} & -\sqrt{1-p} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{1-p} \\ \sqrt{p} \end{pmatrix} = \sqrt{1-p}|0\rangle + \sqrt{p}|1\rangle.$$

Algoritmo 7: Treinamento rede qPLN

```

1 Inicialize todas as posições de memória com o estado  $|u\rangle$ ;
2 enquanto Critério de parada não for alcançado faça
3   Apresente um padrão  $|p\rangle$  do conjunto de treinamento à rede;
4   para  $t=1$  até  $\eta$  faça
5     Permita que a rede produza a saída  $|s\rangle$  para o padrão  $p$ 
6     se  $|s\rangle$  for igual a resposta desejada para o padrão  $|p\rangle$  então
7        $aprender \leftarrow verdadeiro$ 
8       break
9     fim
10  fim
11  se  $aprender$  então
12    Altere os seletores acessados para que o neurônio produza a saída atual
13  senão
14    Altere os seletores acessados para que o neurônio produza o estado  $|u\rangle$ 
15  fim
16 fim

```

O operador U_p pode ser utilizado para definir o neurônio qMPLN, que pode armazenar um conjunto finito de probabilidades p no intervalo $[0, 1]$. Por exemplo, se a probabilidade de obtermos 1 em uma dada posição de um neurônio MPLN é 0,25, então:

$$U_{0,25}|0\rangle = \begin{pmatrix} \sqrt{0,75} & \sqrt{0,25} \\ \sqrt{0,25} & -\sqrt{0,75} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{0,75} \\ \sqrt{0,25} \end{pmatrix} = 0,866|0\rangle + 0,5|1\rangle.$$

logo a medição de $U_{0,25}$ retorna $|1\rangle$ com probabilidade 25%, como esperado.

Se o número de possíveis probabilidades armazenadas é p_n , então a matriz A do neurônio qMPLN terá a forma descrita na equação (7.4).

$$A = \begin{pmatrix} U_{p_1} & 0 & \cdots & 0 \\ 0 & U_{p_2} & \ddots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & U_{p_n} \end{pmatrix} \quad (7.4)$$

Assim como no qPLN podemos representar o operador A algebricamente utilizando a representação de produto externo, onde $A = \sum_{i=1}^n |i\rangle \langle i| \otimes U_{p_i}$.

O algoritmo 8 descreve o treinamento de redes qMPLN piramidais baseado no algoritmo clássico. Note que todas as operações são realizadas em qubits da base computacional e portanto este algoritmo não explora propriedades da mecânica quântica.

Algoritmo 8: Treinamento rede qMPLN

```

1  Todos os seletores são inicializados para produzir  $|u\rangle$ ;
2  enquanto Critério de parada não for alcançado faça
3      Apresente um padrão  $|p\rangle$  do conjunto de treinamento à rede;
4      para  $t=1$  até  $\eta$  faça
5          Permita que a rede produza a saída  $|s\rangle$  para o padrão  $|p\rangle$ 
6          se  $|s\rangle$  for igual a resposta desejada para o padrão  $|p\rangle$  então
7               $aprender \leftarrow verdadeiro$ 
8              break
9          fim
10     fim
11     se  $aprender$  então
12         todos os seletores das posições de memória acessadas são redefinidos da seguinte
            forma  $S_1 S_2 = S_1 S_2 + g(r)$  (reforço)
13     senão
14         todos os seletores das posições de memória acessadas são redefinidos da seguinte
            forma:  $S_1 S_2 = S_1 S_2 - g(r)$ . (punição)
15     fim
16      $aprender \leftarrow falso$ 
17 fim
  
```

7.3 qRAM

Nas seções anteriores foram definidos modelos de redes neurais sem peso quânticas, mas seus algoritmos de treinamento não apresentam vantagens quando comparados aos algoritmos clássicos. Nesta seção será definido e analisado um modelo de rede neural sem peso quântica baseado no neurônio RAM e um algoritmo de aprendizado que apresentam vantagens quando comparados aos modelos clássicos.

O neurônio RAM armazena um bit em cada uma das posições de sua memória. Assim como no caso clássico o neurônio qRAM irá armazenar apenas um qubit para cada possível entrada do neurônio. Para isto basta definir a matriz A do neurônio qRAM como na equação 7.5, note que este operador A é o operador não controlado, mas a notação A será mantida para manter a

notação padrão para todos os neurônios quânticos sem peso.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (7.5)$$

A matriz \mathbf{A} da equação 7.5 define um operador quântico sobre dois qubits que aplica o operador \mathbf{X} ao segundo qubit se o primeiro está no estado $|1\rangle$ e atua como o operador identidade se o primeiro qubit está no estado $|0\rangle$.

A figura 7.4 descreve o circuito quântico de um neurônio qRAM com duas entradas $|\psi\rangle$ e $|\phi\rangle$, quatro seletores $|s_i\rangle$ e quatro operadores \mathbf{A}_i iguais ao operador \mathbf{A} , onde o primeiro qubit é o seletor $|s_i\rangle$ e o segundo é o estado do registro de saída $|o\rangle$. Pode se verificar facilmente que o aprendizado do neurônio qRAM é obtido pela adaptação dos valores dos seletores de acordo com o conjunto de treinamento.

Quando os seletores do neurônio qRAM estão na base computacional ele se comporta exatamente como um neurônio RAM. Através de um exemplo será verificado o que ocorre com um neurônio qRAM \mathbf{N} quando ele recebe valores superpostos em seus registradores de entrada. Suponha que $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, $|\phi\rangle = 0$, $|s_1 s_2 s_3 s_4\rangle = |0111\rangle$ e $|o\rangle = 0$.

A equação (7.6) mostra a operação do neurônio qRAM. Pela linearidade dos operadores quânticos o neurônio irá calcular a saída da rede para os dois valores de entrada simultaneamente. Neste exemplo duas matrizes \mathbf{A}_i foram endereçadas e o registro de saída está no estado $|o\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

$$\begin{aligned} \mathbf{N} \left[\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |0\rangle |0111\rangle |0\rangle \right] &= \\ \frac{1}{\sqrt{2}} [\mathbf{N}(|0\rangle |0\rangle |0111\rangle |0\rangle) + \mathbf{N}(|1\rangle |0\rangle |0111\rangle |0\rangle)] &= \\ \frac{1}{\sqrt{2}} [(|0\rangle |0\rangle |0111\rangle |0\rangle) + (|1\rangle |0\rangle |0111\rangle |1\rangle)] & \end{aligned} \quad (7.6)$$

Seja $|u\rangle = H|0\rangle$ um estado indefinido como no Capítulo 5. Na equação (7.6) o neurônio qRAM se comporta como um neurônio GSN. O neurônio pode receber e produzir $|0\rangle$, $|1\rangle$ e $|u\rangle$ então o neurônio qRAM pode ser visto como um neurônio qGSN, mas o neurônio qRAM pode

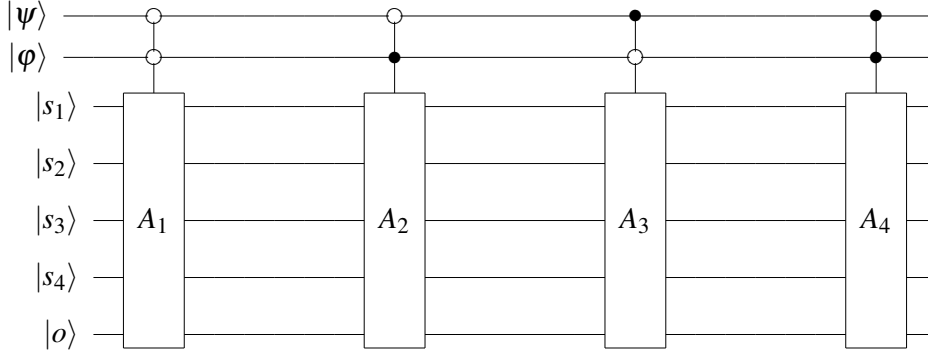


Figura 7.4 qRAM Node

receber e produzir outros estados quânticos se comportando como uma espécie de neurônio GSN com valores contínuos.

7.3.1 Simulação dos modelos de redes neurais sem peso clássicos

Se uma medição do registrador de saída do nodo qRAM for realizada e seletores em superposição forem permitidos, o neurônio qRAM pode ser utilizado para simular os neurônios PLN, MPLN e pRAM e os algoritmos de treinamento das redes neurais sem peso clássicas podem ser adaptados ao neurônio qRAM.

O algoritmo clássico da rede PLN é definido no Algoritmo 4. Os valores armazenados na memória de um neurônio PLN são 0, 1 e u , estes valores serão representados respectivamente pelos qubits $|0\rangle$, $|1\rangle$ e $|u\rangle = H|0\rangle$. A saída probabilística do neurônio PLN é obtida através de uma medição da saída do neurônio qRAM. O resultado desta medição é descrito na equação (7.7).

$$y = \begin{cases} 0 & \text{if } |o\rangle = |0\rangle \\ 1 & \text{if } |o\rangle = |1\rangle \\ \text{random}(0,1) & \text{if } |o\rangle = |u\rangle \end{cases} \quad (7.7)$$

Com uma pequena modificação do algoritmo 4 o neurônio qRAM pode ser treinado. Para isto é necessário medir a saída de cada neurônio, definir $|u\rangle = H|0\rangle$, substituir a linha 1 do Algoritmo 4 por *inicializar todos seletores com o estado quântico $|u\rangle$* e substituir a linha 13 e 15 respectivamente por *alterar todos os seletores dos operadores A endereçados para produzir a saída atual* e *alterar todos os seletores dos operadores A endereçados para produzir $|u\rangle$* .

O neurônio PLN é um caso particular do neurônio qRAM com uma medição realizada no registrador de saída.

Nos neurônios qPLN e qMPLN o tamanho das matrizes A depende do número de probabilidades que podem ser armazenadas, logo a quantização de neurônios pRAMs possui um custo elevado. Por outro lado, no neurônio qRAM as matrizes A do neurônio qRAM independem do número de probabilidades a serem armazenadas. As probabilidades são armazenadas como amplitudes de probabilidades dos qubits utilizados como seletores, que podem ser ajustadas durante o aprendizado utilizando rotações de fase.

O neurônio qRAM pode ser utilizado para simular os modelos de redes neurais sem peso clássicas. Isto nos mostra que qualquer computação realizada por uma rede neural sem peso pode ser realizada por um neurônio qRAM, o que implica, por exemplo, que o neurônio qRAM mantém as propriedades teóricas e práticas dos modelos de redes neurais sem peso. Na próxima seção um algoritmo de treinamento quântico para o treinamento de nodos qRAM será apresentado.

7.3.2 Algoritmo de Aprendizado Baseado na Superposição (AAS)

O AAS é um algoritmo supervisionado que pode ser utilizado para treinar redes qRAM onde os seletores e entradas podem ser preparados como um estado quântico $|\psi\rangle$ com quatro registradores quânticos. O primeiro registrador s irá armazenar os seletores, o segundo registrador p irá armazenar os padrões de treinamento, o terceiro o com um qubit é reservado para a saída do neurônio e o quarto registrador d é utilizado apenas durante o treinamento e irá armazenar as saídas desejadas.

A idéia básica do algoritmos AAS é criar uma superposição de todos as possíveis combinações dos valores para os quatro registradores do estado $|\psi\rangle$. O estado $|\psi\rangle$ é apresentado ao neurônio que calcula a saída para cada todos os seletores em superposição no registrador s simultaneamente. Em seguida o algoritmo de Grover é aplicado sobre o registrador s buscando o conjunto de seletores em que os registradores o e d estão no mesmo estado. Uma medição no registrador s irá retornar os parâmetros desejados com alta probabilidade.

No primeiro passo do algoritmo 9 uma superposição de todos as possíveis configurações da rede é criada. Esta passo é realizado inicializando o registrador quântico s com o estado $|0, \dots, 0\rangle$ e aplicando o operador Hadamard em todos os qubits do registrador s . Neste momento, s irá armazenar o estado quântico $|s\rangle$ descrito na equação (7.8), onde m é a quantidade de seletores,

Algoritmo 9: Aprendizado Baseado na Superposição

-
- 1 Inicialize todos os qubits no registro **s** com o estado quântico $H|0\rangle$.
 - 2 Inicialize o registro **p** com o estado quântico $|p\rangle = \sum_{i=1}^P |p_i, 0, d_i\rangle$.
 - 3 $|\psi\rangle = N|\psi\rangle$, onde N é um operador quântico representando a ação do neurônio.
 - 4 Aplique uma operação de inversão de fase no estado $|\psi\rangle$ onde os registros $\mathbf{p}, \mathbf{o}, \mathbf{d} = \sum_{i=1}^P |p_i, d_i, d_i\rangle$
 - 5 $|\psi\rangle = N^{-1}|\psi\rangle$, para desemaranhar o estado no registrador **s**
 - 6 Aplique o operador de inversão sobre a média $-I + 2A$ no registrador **s**
 - 7 Repita os passos 3, 4, 5 e 6 $T = \frac{P}{4}\sqrt{n}$ vezes, onde n é o número de seletores possíveis para a rede.
 - 8 Realize uma medição no registrador **s** para obter os parâmetros desejados.
-

$a = 2^m - 1$ and $s_j = j$.

$$|s\rangle = \frac{1}{\sqrt{a+1}} \sum_{j=0}^a |s_j\rangle \quad (7.8)$$

Denotando um padrão do conjunto de treinamento por p_i e sua resposta desejada por d_i , no segundo passo do algoritmo um estado quântico superposto armazenando todos p_i e d_i é armazenado no registrador quântico **p** e **d** respectivamente. O registrador de saída **o** é inicializado com o estado da base computacional $|0\rangle$. Este passo pode ser realizado utilizando o algoritmo de inicialização da memória quântica probabilística descrito no capítulo 4. Neste momento o o registrador **p** armazena todos padrões de treinamento p_i e o registro **d** armazena todas as respostas desejadas d_i . O estado dos registradores **p**, **o** e **d** é descrito na equação (7.9).

$$|p, o, d\rangle = \frac{1}{\sqrt{P}} \sum_{i=0}^{P-1} |p_i, 0, d_i\rangle \quad (7.9)$$

Seja N o operador quântico associado a uma rede neural quântica. A ação de N pode ser descrita como enviando o estado $|s, p_i, 0, d_i\rangle$ para o estado $|s, p_i, y_i, d_i\rangle$, onde y_i é a saída calculada do neurônio.

No terceiro passo do algoritmo 9 o operador N age no estado $|\psi_0\rangle$ e, pela linearidade dos operadores quânticos, a saída desejada é produzida em todas as parcelas da superposição. A equação (7.10) descreve o estado $|\psi_1\rangle$, onde m é a cardinalidade do conjunto de treinamento e $a = 2^m - 1$.

$$|\psi\rangle = \frac{1}{\sqrt{a+1}} \frac{1}{\sqrt{p}} \sum_{i=0}^{p-1} \sum_{j=0}^a |s_j, p_i, y_i, d_i\rangle \quad (7.10)$$

Seja U_f um operador quântico que envia o estado $\sum_{i=0}^{p-1} |p_i, d_i, d_i\rangle$ para $-\sum_{i=0}^{p-1} |p_i, d_i, d_i\rangle$ e que age como o operador identidade para os outros estados. No quarto passo do algoritmo 9 o operador $U_f(I \otimes H)$ recebe o estado $|\psi_1\rangle$ e irá marcar apenas as parcelas onde as respostas desejadas são iguais a saída calculada $y_i = d_i$.

No terceiro passo N emaranha o registrador quântico s com os registradores quânticos, p , o e d , então o algoritmo quântico de busca aplicado no registrador s irá alterar os registradores p , o e d . Para evitar esta alteração nos registros p , o e d o operador N^{-1} é aplicado para desemaranhar os estados nos registradores no quinto passo do algoritmo.

No sexto passo do algoritmo a operação de inversão sobre a média é aplicada aumentando a amplitude de probabilidade dos estados marcados pela inversão de fase no quarto passo do algoritmo. Os passos 3, 4, 5 e 6 devem ser repetidos T vezes e uma medição do registrador s irá retornar os parâmetros desejados.

Com uma pequena alteração do algoritmo 9 pode ser definido um algoritmo onde a rede neural precisa ser executada apenas duas vezes, independente do tamanho do conjunto de treinamento. Note que assintoticamente esta mudança não irá alterar o custo computacional do algoritmo, pois a busca quântica terá o mesmo número de passos que no algoritmo 9.

O algoritmo 10 difere do algoritmo 9 por ter mais um registrador e que será utilizado para marcar os parâmetros que fazem a rede produzir a resposta desejada para todos os padrões do conjunto de treinamento. A operação de inversão de fase será baseada no registrador e e será necessário executar a rede apenas duas vezes.

7.3.3 Complexidade do algoritmo AAS

Para simplificar a análise do algoritmo suponha que exista um único conjunto de seletores que se ajusta ao conjunto de treinamento, todos os possíveis padrões estejam no conjunto de treinamento e que o tamanho do conjunto de treinamento é $N_p = 2^n$. Nesta situação os padrões podem ser representados por $n = \log N_p$ bits. Uma rede qRAM piramidal com $\log N_p$ entradas terá $2^{\log(\log(N_p-1))}$ neurônios e a mudança de fase no processo de busca irá marcar $2^{\log(N_p)}$ parcelas da superposição. Então o número de qubits no registro quântico s será $4(\log N_p - 1)$. A

Algoritmo 10: Aprendizado Baseado na Superposição de Única Apresentação

- 1 Inicialize todos os qubits no registro \mathbf{s} com o estado quântico $H|0\rangle$.
 - 2 Inicialize o registro \mathbf{p} com o estado quântico $|p\rangle = \sum_{i=1}^P |p_i, 0, d_i\rangle$.
 - 3 $|\psi\rangle = N|\Psi\rangle$, onde N é um operador quântico representando a ação do neurônio.
 - 4 Faça o estado do registrador $\mathbf{e} |e\rangle = |1\rangle$ onde os registros $\mathbf{p}, \mathbf{o}, \mathbf{d} = \sum_{i=1}^P |p_i, d_i, d_i\rangle$
 - 5 $|\Psi\rangle = N^{-1}|\psi\rangle$, para desemaranhar o estado no registrador \mathbf{s}
 - 6 Aplique uma operação de inversão de fase no estado $|\Psi\rangle$ onde o registrador $\mathbf{e} = |1\rangle$
 - 7 Aplique o operador de inversão sobre a média $-I + 2A$ no registrador \mathbf{s}
 - 8 Repita os passos 6 e 7 $T = \frac{P}{4}\sqrt{n}$ vezes, onde n é o número de seletores possíveis para a rede.
 - 9 Realize uma medição no registrador \mathbf{s} para obter os parâmetros desejados.
-

busca ocorrerá no registrador quântico \mathbf{s} com todos os $2^{4(\log N_p - 1)}$ possíveis seletores. O custo desta busca será $\left\lfloor \frac{\pi}{4} \sqrt{\frac{2^{4(\log N_p - 1)}}{2^{\log(N_p)}}} \right\rfloor = \left\lfloor \frac{\pi}{4} \left(\frac{2^{2(\log N_p - 1)}}{\sqrt{N_p}} \right) \right\rfloor = \left\lfloor \frac{\pi}{4} \left(2^{-2} \cdot N_p^{1,5} \right) \right\rfloor = \left\lfloor \frac{\pi}{16} \left(N_p^{1,5} \right) \right\rfloor$.

Com o resultado do parágrafo anterior podemos enunciar o Teorema 7.

Teorema 7. *Sejam N_p o tamanho do conjunto de treinamento e n o número de entradas de uma rede neural qRAM com arquitetura e piramidal, onde cada neurônio possui duas entradas. Se $N_p \geq 2^n$, então o algoritmo SLA possui custo polinomial em relação ao tamanho do conjunto de treinamento.*

7.3.4 Treinamento do neurônio qRAM com o AAS

Nesta seção o SLA será utilizado para o treinamento do neurônio qRAM para resolver o problema da paridade de 2 bits. Para este problema um neurônio qRAM com duas entradas aprende a função com uma única iteração. O conjunto de treinamento escolhido é $S = \{(00, 0), (01, 1), (10, 1), (11, 1)\}$.

Um neurônio qRAM com duas entradas tem quatro seletores, então no primeiro passo o estado $|s\rangle$ é inicializado como na equação 7.11.

$$|s\rangle = H^{\otimes} |0000\rangle = \frac{1}{4} \sum_{i=0}^{16} |i\rangle_4 \quad (7.11)$$

No segundo passo os registradores \mathbf{p} , \mathbf{o} e \mathbf{d} são preparados como na equação (7.12).

$$|p, o, d\rangle = \frac{1}{2} (|00, 0, 0\rangle + |01, 0, 1\rangle + |10, 0, 1\rangle + |11, 0, 1\rangle) \quad (7.12)$$

Neste momento o estado $|\psi\rangle$ pode ser descrito com na equação (7.13).

$$|\psi\rangle = \frac{1}{8} \sum_{i=0}^{16} |i\rangle_4 (|00, 0, 0\rangle + |01, 0, 1\rangle + |10, 0, 1\rangle + |11, 0, 1\rangle) \quad (7.13)$$

Em seguida o operador N é aplicado ao estado $|\psi\rangle$ e as saídas y_p^i do neurônio são calculadas para cada seletor i e para cada padrão p como na equação 7.14.

$$|\psi\rangle = \frac{1}{8} \sum_{i=0}^{16} |i\rangle_4 (|00, y_{00}^i, 0\rangle + |01, y_{01}^i, 1\rangle + |10, y_{10}^i, 1\rangle + |11, y_{11}^i, 1\rangle) \quad (7.14)$$

No quarto passo é realizada uma inversão de fase do estado $|e_1\rangle = (|00, 0, 0\rangle + |01, 1, 1\rangle + |10, 1, 1\rangle + |11, 0, 0\rangle)$ ao qubit $|\psi\rangle$ e é obtido o estado descrito na equação (7.15), onde δ é o delta de Kronecker's e $x(i) = 1$ se e somente se $|p_i, o_i, d_i\rangle = |e_1\rangle$. A inversão de fase marca o estado com os parâmetros desejados.

$$|\psi\rangle = \frac{1}{8} \sum_{i=0}^{16} (-1)^{\delta_{1x(i)}} |i\rangle_4 (|00, y_{00}^i, 0\rangle + |01, y_{01}^i, 1\rangle + |10, y_{10}^i, 1\rangle + |11, y_{11}^i, 1\rangle) \quad (7.15)$$

Em seguida o operador N^{-1} é aplicado em $|\psi\rangle$ para desemaranhar os registradores e o operador de inversão sobre a média é aplicado sobre o registrador \mathbf{s} . Neste exemplo, o somatório da amplitude dos 4 estados marcados ao quadrado irá para um enquanto a amplitude dos outros estados irá para zero. O treinamento é finalizado, pois $\left[T = \frac{p_i}{16} \cdot 4^{1,5} \right] = 1$, então o registrador \mathbf{s} pode ser medido e o resultado pode ser utilizado como os parâmetros livres da rede.

7.4 Implementando um conjunto universal de operadores

Os operadores Toffoli e Hadamard formam um conjunto universal de operadores para a computação quântica (veja definição 18), como foi mostrado por Aharonov [2] e Shi [48].

Se um neurônio qPLN ou qMPLN com entradas $|\psi\rangle$ e $|\phi\rangle$ produzir a saída $H|\phi\rangle$, o teorema 5 será violado, pois a ação do neurônio irá levar o estado $|\psi\rangle|\phi\rangle|0\rangle$ em $|\psi\rangle|\phi\rangle H|\phi\rangle$ e aplicando o operador $I \otimes I \otimes H$ será obtido $|\psi\rangle|\phi\rangle|\phi\rangle$. Pelo teorema da não-clonagem de estados quânticos chega-se a seguinte conclusão:

Lema 3. *Neurônios qPLN e qMPLN com o registro de saída inicializado com um estado fixo $|0\rangle$ não pode implementar o operador Hadamard para um qubit arbitrário $|\phi\rangle$. \square*

Definição 17. (Universalidade Estrita) *Um conjunto de portas quânticas S é estritamente universal se existem uma constante n_0 tal que para qualquer $n \geq n_0$, o subgrupo gerado por S é denso em $SU(2^n)$, o subgrupo das matrizes unitárias com determinante 1 operando sobre n qubits.*

Definição 18. (Universalidade computacional) *Um conjunto de portas quânticas C é computacionalmente universal se puder ser utilizado para simular com erro ε qualquer circuito quântico que utiliza n qubits e t operadores de um conjunto estritamente universal com custo logarítmico em $(n, t, 1/\varepsilon)$.*

Para permitir que o operador Hadamard possa ser implementado por neurônios qPLN e qMPLN é necessário permitir que o registrador de saída possa ser inicializado com um estado quântico arbitrário. Isto irá alterar o comportamento dos neurônios. Por exemplo, inicializar o registro de saída com o estado $|1\rangle$ irá inverter o comportamento do neurônio, o neurônio irá ter como saída 1 quando 0 for esperado e 0 quando 1 for esperado. Quando o registrador de saída for inicializado com um estado em superposição os neurônios não terão um comportamento similar a sua contraparte clássica.

Teorema 8. *Circuitos compostos por neurônios qPLN ou qMPLN com registrador de saída inicializável podem aproximar qualquer função quântica..*

Será utilizado, sem perda de generalidade, um neurônio com 3 entradas para simular os operadores Hadamard e Toffoli. Para isto, pelo lema 3, será necessário considerar que o neurônio qPLN ou qMPLN possuem registrador de saída inicializável e que o número de saídas

no neurônio é ajustável, i.e, 1,2 ou todas as linhas do circuito do neurônio podem ser utilizadas como saída.

As configurações dos neurônios para simular os operadores quânticos Hadamard e Toffoli são apresentadas na figura 7.5 e na figura 7.6, respectivamente. Nas figuras X , I ou U significam que a matriz A está recebendo seletores que fazem que ela tenha a ação dos operadores quânticos X , I ou U . Onde U é qualquer operador possível para o neurônio. \square

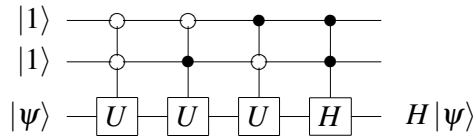


Figura 7.5 NLQ simulando Hadamard

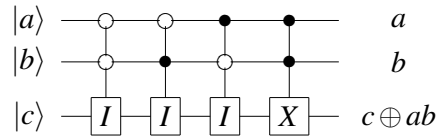


Figura 7.6 NLQ simulando operador Toffoli

O teorema 8 mostra que os neurônios qPLN e qMPLN podem ser utilizados para aproximar qualquer circuito quântico. No entanto, a utilização do registrador de saída com uma das entradas do neurônio requer um estudo mais detalhado.

7.5 Sumário do capítulo

O principal resultado obtido neste capítulo foi o de que as redes neurais sem peso possuem equivalentes quânticos que podem ser definidos de forma natural e que preservam as características dos modelos neurais sem peso, como seus algoritmos de treinamento e topologia. Este resultado permite que a busca por algoritmos quânticos para as redes neurais sem peso seja considerada como uma generalização dos modelos clássicos preservando os resultados teóricos como a equivalência com automatos probabilísticos e máquinas de turing e dos resultados

práticos que mostram que as redes neurais sem peso possuem boa capacidade de generalização, isto é, todo o trabalho no desenvolvimento das redes neurais sem peso pode ser diretamente aplicados aos modelos quânticos propostos.

Quanto ao aprendizado em sistema quânticos foi verificado que o algoritmo de busca quântico pode ser utilizado para o treinamento de redes neurais sem peso, este algoritmo foi denominado algoritmo de aprendizado baseado na superposição. A principal característica apresentada por este algoritmo é a de que todos os padrões podem ser apresentados a rede simultaneamente, o que não pode ser realizado no caso clássico.

O segundo algoritmo definido, denominado algoritmo de aprendizado baseado na superposição de única apresentação é uma alteração do SLA. Neste algoritmo a rede precisa ser executada apenas duas vezes, no entanto o SSSLA possui custo computacional igual ao do SLA. A capacidade de treinar a rede com uma única execução levanta a possibilidade da existência de um algoritmo com custo computacional mais baixo. Para isto, podemos por exemplo substituir o algoritmo de Grover pelo o algoritmo de recuperação da memória quântica probabilística.

A seção 7.4 mostra que os modelos de redes neurais quânticas sem peso não são computacionalmente universais, mas que uma pequena alteração (definida no Lema 3) nos modelos qPLN e qMPLN fazem com que estes sejam computacionalmente universais. O lema 3 sugere a definição de um novo modelo, com registrador de saída inicializável, no entanto o algoritmo de treinamento e arquitetura para tal modelo não seria simples. A alteração do valor inicial do registrador de saída irá alterar o comportamento do neurônio e a existência de tipos diferentes de entrada faz com que a escolha da arquitetura da rede se torne muito mais complexa.

É importante salientar que os modelos de redes neurais com peso não possuem equivalentes quânticos que preservem os algoritmos de treinamento, pois o treinamento destas redes dependem de funções não lineares que dificilmente terão análogas quânticas [6] e do cálculo do gradiente descendente. Como visto no capítulo 6, mesmo para um modelo de rede neural com única camada como o perceptron, os algoritmos de treinamento propostos possuem problemas com o respeito aos postulados da mecânica quântica.

CAPÍTULO 8

Conclusão

Este capítulo é a conclusão desta dissertação. Na seção 8.1 são realizadas algumas considerações finais, na seção 8.2 são descritas as contribuições deste trabalho e na seção 8.3 são descritos os possíveis trabalhos futuros a esta dissertação.

8.1 Considerações finais

O estudo dos modelos quânticos neurais é uma área recente e em expansão. Muitas pesquisas estão sendo realizadas para generalizar modelos de redes neurais aos domínios da computação quântica, por exemplo, o perceptron, perceptron de múltiplas camadas, redes neurais fuzzy e redes neurais complexas. Nesta dissertação foram apresentados modelos de redes neurais quânticas sem peso [39, 37, 51, 53, 52].

O presente trabalho está contextualizado no estudo dos novos modelos de computação, o estudo dos modelos não convencionais de computação é um dos grandes desafios da computação no Brasil [1]. O foco principal desta dissertação foi o desenvolvimento de algoritmos de treinamento para redes neurais quânticas sem peso utilizando como base uma memória quântica probabilística. O desenvolvimento de algoritmos de aprendizagem quânticos, podem permitir a utilização da computação quântica em uma maior gama de problemas.

Neste trabalho foram apresentadas quantizações de vários modelos de redes neurais, como a do nodo RAM, nodo lógico probabilístico (PLN) e do nodo lógico probabilístico multivalorado (MPLN), assim como os algoritmos de treinamento quânticos utilizando baseados no princípio da superposição.

8.2 Contribuições deste trabalho

Algumas das contribuições deste trabalho foram:

(i) O desenvolvimento de modelos de redes neurais quânticas sem pesos (qRAM, qPLN e qMPLN), permitindo o estudo dos modelos neurais sem peso utilizando os princípios da computação quântica. A quantização dos modelos de redes neurais com pesos não é um processo simples, pois requer a discretização dos parâmetros, logo algoritmos de treinamento baseados na continuidade e diferenciabilidade das funções não serão facilmente quantizados. Os algoritmos de treinamento das redes neurais sem peso não são baseados nestas técnicas e podem ser facilmente quantizados como mostrado no capítulo 7.

(ii) A elaboração de algoritmos de aprendizado quânticos para redes neurais sem peso. Em particular o algoritmo de aprendizado baseado na superposição de única apresentação (SSSLA), onde a rede neural é executada apenas duas vezes durante o treinamento. Apesar de não ter ganho computacional em relação ao algoritmo de aprendizado baseado na superposição o SSSLA mostra que talvez exista a possibilidade de redução do custo computacional do algoritmo de aprendizado, pois podemos treinar a rede com um número fixo de execuções, independente do tamanho do conjunto de treinamento e da representação da rede neural.

(iii) A análise dos modelos propostos mostram que os neurônios qRAM, qPLN e qMPLN não são universais, como foi verificado no Lema 3. No entanto uma pequena modificação nos neurônios qPLN e qMPLN fazem com que redes (ou circuitos) compostos por neurônios qPLN e qMPLN possam aproximar qualquer circuito quântico. Para isto foi necessário tornar inicializável o registrador de saída dos neurônios.

8.3 Trabalhos Futuros

Os algoritmos de aprendizado propostos nesta dissertação podem ser vistos como uma aplicação da memória quântica probabilística proposta por Ventura [44] que basicamente consiste no algoritmo de armazenamento descrito no capítulo 4 e no uso do algoritmo de busca de Grover [15] para a recuperação da informação. Existem outras possibilidades para a recuperação de informação em uma memória associativa quântica, como a proposta por Trugenberger e descrita na seção 4.3. A análise do desempenho dos algoritmos de treinamento para redes neurais baseados na superposição substituindo o algoritmo de Grover pelo algoritmo de recuperação da memória quântica probabilística proposto por Trugenberger é um dos possíveis trabalhos futuros a esta dissertação.

Outra possibilidade é o desenvolvimento de um algoritmo de aprendizado para os neurônios

qPLN e qMPLN onde o registrador de entrada não seja fixo. Tal algoritmo teria que ser dividido em duas fases i) escolha da arquitetura da rede, pois os neurônios passariam a ter entradas de dois diferentes tipos; e ii) atualização dos seletores. Como visto na seção 7.4 sem uma entrada fixa os neurônios qPLN e qMPLN podem aproximar qualquer circuito quântico.

A análise teórica dos modelos e o estudo de seu poder computacional podem ser estendidos verificando se as redes neurais quânticas sem peso podem ser utilizadas para simular autômatos e máquinas de Turing quânticas. Mais especificamente pode ser verificado que linguagens podem ser reconhecidas por redes neurais quânticas sem pesos com arquitetura piramidal.

A simulação de sistemas quânticos possui custo exponencial, o que torna inviável a realização de experimentos com redes neurais quânticas em problemas reais com alta dimensionalidade. Um possível trabalho futuro é a realização de experimentos para comparar os modelos quânticos neurais e as redes neurais clássicas em problemas com baixa dimensão.

Referências Bibliográficas

- [1] Grandes desafios da pesquisa em computação no brasil – 2006 – 2016. Relatório, Sociedade Brasileira de Computação, 2006. http://www.ic.unicamp.br/~cmbm/desafios_SBC/RelatorioFinal.pdf. Acessado em: 20/01/2011.
- [2] D. Aharonov. A simple proof that toffoli and hadamard are quantum universal. *arXiv:quant-ph/0301040*, 2003.
- [3] R. Al-Alawi. Fpga implementation of a pyramidal weightless neural networks learning system. *International Journal of Neural Systems*, 13(4):225–237, 2003.
- [4] I. Aleksander. Self-adaptive universal logic circuits. *Electronics Letters*, 2(8):321–322, 1966.
- [5] I. Aleksander, M. de Gregorio, F. M. G. França, P. Lima, and H. Morton. A brief introduction to weightless neural systems. In *European Symposium on Artificial Neural Networks, 2009. Proceedings.*, pages 299–305, Bruges, Belgium, 2009.
- [6] M. V. Altaisky. Quantum neural network. Technical report, Joint Institute for Nuclear Research, Russia, 2001.
- [7] R. Bowmaker and G. Coghill. Improved recognition capabilities for goal seeking neuron. *Electronics Letters*, 28(3):220 – 221, 1992.
- [8] A. P. Braga, A. P. Carvalho, and T. B. Ludermit. *Redes Neurais Artificiais: Teoria e Aplicações*. LTC, 2000.
- [9] A. Carvalho, D. Bisset, and M. Fairhurst. Training algorithms for gsnf neural networks. In *Proceedings of the Second Workshop on Cybernetic Vision*, pages 74–79, São Carlos, 1996. IEEE Computer Press.
- [10] E. C. B. Carvalho Filho, M. C. Fairhurst, and D. L. Bisset. A goal seeking neuron for boolean neural networks. In *Proceedings of the International Neural Network Conference*, pages 894–897. IEEE, 1990.

- [11] M.C.P. de Souto. *Computability and Learnability in Sequential Weightless Neural Networks*. PhD thesis, Imperial College of Science, 1999.
- [12] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London Ser. A*, A400:97–117, 1985.
- [13] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467, 1982.
- [14] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.
- [15] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, Jul 1997.
- [16] S. Haykin. *Neural Networks*. Prentice Hall, 1999.
- [17] M. Hirvensalo. *Quantum computing*. Springer-Verlag New York, Inc., New York, NY, USA, 2003.
- [18] K. Huarng and T. H. Yu. The application of neural networks to forecast fuzzy time series. *Physica A: Statistical Mechanics and its Applications*, 363(2):481 – 491, 2006.
- [19] S. C. Kak. On quantum neural computing. *Information Sciences*, 83(3 – 4):143–160, 1995.
- [20] W. K. Kan and I. Aleksander. A probabilistic logic neuron network for associative learning. *IEEE First International Conference on Neural Networks (ICNN87), volume II, pages 541-548*, 2:541–548, 1987.
- [21] N. Kasabov. To spike or not to spike: A probabilistic spiking neuron model. *Neural Networks*, 23(1):16 – 19, 2010.
- [22] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Boston, MA, USA, 2002.
- [23] N. Kouda, N. Matsui, H. Nishimura, and F. Peper. Qubit neural network and its learning efficiency. *Neural Comput. Appl.*, 14(2):114–121, 2005.

- [24] V. M. Krasnopolsky and M. S. Fox-Rabinovitz. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2):122 – 134, 2006.
- [25] E. L. Lima. *Álgebra Linear*. Instituto de Matemática Pura e Aplicada, 1995.
- [26] P. J. Lisboa and A. F. G. Taktak. The use of artificial neural networks in decision support in cancer: A systematic review. *Neural Networks*, 19(4):408 – 415, 2006.
- [27] T. B. Ludermir, A. Carvalho, A. P. Braga, and M. C. P. Souto. Weightless neural models: A review of current and past works. *Neural Computing Surveys*, 2:41 – 61, 1999.
- [28] W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659 – 1671, 1997.
- [29] R. Majhi, G. Panda, and G. Sahoo. Efficient prediction of exchange rates with low complexity artificial neural network models. *Expert Systems with Applications*, 36(1):181 – 189, 2009.
- [30] W. Martins and Jr. Pinheiro, C. G. On-line expansion of goal seeking neuron networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, Como, Italy, 2000. IEEE-INNS-ENNS.
- [31] W. S McCulloch and W. Pitts. A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133, 1930.
- [32] T. Menneer and A. Narayanan. Quantum-inspired neural networks. Technical Report R329, Department of Computer Science, University of Exeter, Exeter, United Kingdom, 1995.
- [33] M. Minsky and S. Papert. *Perceptrons*. The MIT Press, 1969.
- [34] C. Monroe. Quantum information processing with atoms and photons. *Nature*, 416:238 – 246, 2002.
- [35] M. A. Nielsen and I. L. Chuang. Programmable quantum gate arrays. *Physical Review Letters*, 2, 1997.
- [36] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

- [37] W. R. Oliveira. Quantum ram based neural networks. In *European Symposium on Artificial Neural Networks, 2009. Proceedings.*, pages 331–336, April 2009.
- [38] W. R. Oliveira, W. Galindo, A. Leonel, J. Pereira, and A. J. Silva. Redes neurais quânticas sem peso. In *2º Workshop-Escola em Computação e Informação Quântica*, pages 133–139, Campina Grande, Paraíba, 2007. EDUEFCG.
- [39] W. R. Oliveira, A. J. Silva, T.B. Ludermir, W. Galindo, A. Leonel, and J. Pereira. Quantum logical neural networks. In *12th Brazilian Symposium on Neural Networks, 2010*, pages 147 – 152, Salvador, Bahia, 2008. IEEE.
- [40] M. Panella and G. Martinelli. Neurofuzzy networks with nonlinear quantum learning. *Fuzzy Systems, IEEE Transactions on*, 17(3):698 –710, june 2009.
- [41] M. Panella and G. Martinelli. *International Journal of Circuit Theory & Applications*, to be published. doi: 10.1002/cta.619.
- [42] H. Pao. Forecasting electricity market pricing using artificial neural networks. *Energy Conversion and Management*, 48(3):907 – 912, 2007.
- [43] R. Penrose. *The Emperor’s New Mind*. Oxford University Press, 1989.
- [44] B. Ricks and D. Ventura. Training a quantum neural network. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
- [45] R. Rohwer and M. Morciniec. The theoretical and experimental status of the n-tuple classifier. *Neural Networks*, 11(1):1 – 14, 1998.
- [46] B. Ruf and M. Schmitt. Self-organization of spiking neurons using action potential timing. *Neural Networks, IEEE Transactions on*, 9(3):575 –578, May 1998.
- [47] R. Shankar. *Principles of Quantum Mechanics*. Plenum Press, 1994.
- [48] Y. Shi. Both toffoli and controlled-not need little help to do universal quantum computation. *Quantum Information and Computation*, 3(1):84–92, 2003.
- [49] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Computing*, 26:1484 – 1509, 1997.
- [50] A. J. Silva, N. L. Mineu, and T. B. Ludermir. Evolving artificial neural networks using adaptive differential evolution. In *IBERAMIA 2010, LNAI 6433*, pages 396–405, 2010.

- [51] A. J. Silva, W. R. Oliveira, and T. B. Ludermir. Superposition based learning algorithm. In *11th Brazilian Symposium on Neural Networks, 2010*, pages 1–6, São Bernardo do Campo, 2010. IEEE Computer Press.
- [52] A. J. Silva, W. R. Oliveira, and T. B. Ludermir. Um algoritmo de aprendizado probabilístico para redes neurais quânticas baseado na superposição. In *III Workshop-Escola em Computação e Informação Quântica*, pages 48 – 57, Petrópolis, Rio de Janeiro, 2010. Laboratório Nacional de Computação Científica.
- [53] A. J. Silva, W. R. Oliveira, and T. B. Ludermir. A weightless neural node based on a probabilistic quantum memory. In *11th Brazilian Symposium on Neural Networks, 2010*, pages 259–264, São Bernardo do Campo, 2010. IEEE Computer Press.
- [54] M. Tegmark. Why the brain is probably not a quantum computer. *Information Sciences*, 128(3 – 4):155–179, 2000.
- [55] C. A. Trugenberger. Probabilistic quantum memories. *Phys. Rev. Lett.*, 87(6):067901, Jul 2001.
- [56] C. A. Trugenberger. Probabilistic quantum memories. *Phys. Rev. Lett.*, 87(6):067901, 2001.
- [57] D. Ventura and T. Martinez. An artificial neuron with quantum mechanical properties. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 482–485, 1997.
- [58] D. Ventura and T. Martinez. Quantum associative memory. *Information Sciences*, 124(1-4):273 – 296, 2000.
- [59] Z. Xu, R. Zhang, and W. Jing. When does online bp training converge? *Neural Networks, IEEE Transactions on*, 20(10):1529–1539, Oct. 2009.
- [60] A. Yamazaki, T. B. Ludermir, and M. C. P. Souto. Classification of vintages of wine by artificial nose using time delay neural networks. *Electronics Letters*, 37(24):1466–1467, 2001.
- [61] C. Zanchettin and T. B. Ludermir. Hybrid neural systems for pattern recognition in artificial noses. *Int. J. Neural Syst.*, 15(1-2):137–149, 2005.
- [62] R. Zhou and Q. Ding. Quantum m-p neural network. *Int J Theor Phys*, 46(12):3209 – 3215, 2007.