



Universidade Federal do Rio Grande do Norte

Centro de Tecnologia - CT

Curso de Engenharia Mecatrônica

Computação quântica - Uma abordagem para a construção de um Perceptron quântico

Samuel Amico Fidelis

Natal, RN

2019

Samuel Amico Fidelis

Computação quântica - Uma abordagem para a construção de um Perceptron quântico

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte como parte dos requisitos para a obtenção do título de Engenheiro Mecatrônico, orientado pelo Prof. Samaheri Moraes Dias e coorientado pelo Prof. Rafael Chaves

Universidade Federal do Rio Grande do Norte (UFRN)

Centro de Tecnologia (CT)

Curso de Engenharia Mecatrônica

Orientador: Samaheri Moraes Dias

Coorientador: Rafael Chaves

Natal, RN

2019

Samuel Amico Fidelis

Computação quântica - Uma abordagem para a construção de um Perceptron quântico

Trabalho aprovado. Natal, RN, 08 de novembro de 2019:

Samaheri Moraes Dias
UFRN - Orientador

Rafael Chaves Souto Araujo
UFRN - Coorientador

Kurios Iuri Pinheiro de Melo Queiroz
UFRN - Avaliador interno

Joilson Batista de Almeida Rego
UFRN - Avaliador interno

Natal, RN
2019

Agradecimentos

Os agradecimentos são direcionados, em especial, a Deus, aos meus pais que sempre me acompanharam e conduziram para no estudo, a meu tio Sandro que me inspirou a seguir na carreira científica, a minha namorada Rayanne Samara pela paciência, amor e carinho, e sem ela o trabalho teria sido recheado de erros de português. Agradeço aos professores Askery Canabarro e Rafael Chaves que me orientaram no entendimento e correções de toda a teoria e experimentos realizados; e ao professor e orientador Samaherni Moraes pelo seu suporte e ajuda no curso; agradeço também ao ilustre cientista ao qual me inspiro, Richard Feynman.

“As pessoas costumam me perguntar o que acontecerá a seguir na ciência que será importante. Claro que o ponto principal é que, se é importante, é algo que não esperávamos”. (Freeman Dyson)

Resumo

O presente trabalho introduz conceitos de computação quântica, informação quântica e aprendizagem de máquina, mostrando os seus significados e onde podem ser aplicados; aborda a teoria de aprendizagem de máquina quântica, que é a junção dos três conceitos que foram introduzidos anteriormente. Em seguida, desenvolve uma aplicação teórica e prática de um algoritmo denominado Perceptron Quântico, que tem por objetivo superar as limitações do seu correspondente clássico que é o Perceptron. Para o experimento é utilizado um conjunto de dados de entrada que são mapeados para suas respectivas saídas de acordo com funções de uma porta lógica binária. Os experimentos são aplicados em um computador quântico teórico e um computador quântico real disponibilizado pela IBM-Q. Ao final, é apresentado o resultado dos experimentos demonstrando a supremacia teórica do novo Perceptron.

Palavras-chave: Computação Quântica. Aprendizagem de Máquina. Perceptron. Aprendizagem de Máquina Quântica.

Abstract

The present work introduces concepts of quantum computing, quantum information and machine learning, showing their meanings and where they can be applied; It addresses the theory of quantum machine learning, which is the junction of the three concepts that were introduced earlier. It then develops a theoretical and practical application of an algorithm called Quantum Perceptron, which aims to overcome the limitations of its classical correspondent, which is Perceptron. For the experiment is used a set of input data that are mapped to their respective outputs according to functions of a binary logic gate. The experiments are applied to a theoretical quantum computer and a real quantum computer made available by IBM-Q. At the end, we present the results of the experiments demonstrating the theoretical supremacy of the new Perceptron.

Keywords: Quantum Computation. Machine learning. Perceptron. Quantum Machine Learning

Lista de ilustrações

Figura 1 – Arquitetura de Von Neumann	15
Figura 2 – Diferença entre Bit e Qubit	17
Figura 3 – Computador Quântico Dwave	18
Figura 4 – Porta quântica NOT e seu funcionamento em um estado $ \psi\rangle$	24
Figura 5 – Porta quântica Z e seu funcionamento em um estado $ \psi\rangle$	25
Figura 6 – Porta quântica Y e seu funcionamento em um estado $ \psi\rangle$	25
Figura 7 – Porta quântica H e seu funcionamento em um estado $ \psi\rangle$	26
Figura 8 – Aplicação de uma porta quântica U em dois qubits	27
Figura 9 – Aplicação de uma porta quântica U em n qubits	27
Figura 10 – Porta quântica CNOT	27
Figura 11 – Porta quântica TOFFOLI	28
Figura 12 – Porta quântica FREDKIN	28
Figura 13 – Classificação de E-mails	30
Figura 14 – Comparação entre os fluxogramas dos algoritmos convencionais e <i>Machine Learning</i>	30
Figura 15 – Perceptron executando computação lógica simples, onde A e B são as entradas e C a saída resultante	32
Figura 16 – Representação gráfica de um Perceptron	32
Figura 17 – Função degrau	33
Figura 18 – Função rampa simétrica	33
Figura 19 – Função sigmoide	34
Figura 20 – Função linear	34
Figura 21 – Representação gráfica da separação linear de dois conjuntos A e B	35
Figura 22 – Diferentes tipos de vertentes que unem aprendizagem de máquina e computação quântica. A diferença entre elas é a forma como o algoritmo é descrito e como o dado é tratado.	37
Figura 23 – Problema linearmente separável e não separável.	39
Figura 24 – Operação da Porta quântica R_ϕ	42
Figura 25 – Circuito quântico para rotacionar e trocar a medição da probabilidade de um qubit de amplitude para fase.	42
Figura 26 – Circuito quântico para função de ativação de dois qubits.	43
Figura 27 – Perceptron Quântico	44
Figura 28 – Porta quântica Hadarmad aplica a um qubit $ 0\rangle$	46
Figura 29 – Circuito quântico <i>feed-forward</i>	46
Figura 30 – Continuação do circuito quântico <i>feed-forward</i>	47
Figura 31 – Representação estendida dos oráculos U_{f1} e U_{f2} juntos.	47

Figura 32 – Circuito do Perceptron Quântico	47
Figura 33 – Circuito do Perceptron Quântico	49
Figura 34 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$	50
Figura 35 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$	51
Figura 36 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$	52
Figura 37 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$	53
Figura 38 – Solução do Algoritmo QPerceptron para a porta OR no simulador da Qskit, onde para as entradas $ 01\rangle, 10\rangle, 11\rangle$	54
Figura 39 – Estados quânticos dos qubits de saída do Algoritmo QPerceptron para a porta OR com as entradas $ 01\rangle$	54
Figura 40 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$	55
Figura 41 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$	56
Figura 42 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$	56
Figura 43 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$	57
Figura 44 – Solução do Algoritmo QPerceptron para a porta AND, onde para a entrada $ 11\rangle$, a probabilidade de se obter o resultado para a saída ser igual a 1 é 100% apenas para essa entrada.	58
Figura 45 – Estados quânticos dos qubits de saída do Algoritmo QPerceptron para a porta OR com as entradas $ 11\rangle$	58
Figura 46 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$	59
Figura 47 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$	60
Figura 48 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$	61
Figura 49 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$	62
Figura 50 – Solução do Algoritmo QPerceptron para a porta XOR, onde para a entrada $ 10\rangle, 01\rangle$, a probabilidade de se obter o resultado para a saída ser igual a 1 é 50% apenas para essa entrada.	63
Figura 51 – Estados quânticos dos qubits de saída do Algoritmo QPerceptron para a porta XOR com as entradas $ 10\rangle$	63
Figura 52 – Arquiteturas do IBM-Q	64
Figura 53 – Circuito do QPerceptron dentro do IBM-Q	65
Figura 54 – Arquitetura IBM-Melbourne	66

Figura 55 – Resultado para a porta AND, utilizando o processador IBM-Melbourne com entrada $ 11\rangle$	66
Figura 56 – Resultado para a porta AND simulado em um processador quântico teórico com entrada $ 11\rangle$	67
Figura 57 – Arquitetura IBM-VIGO	67
Figura 58 – Resultado para a porta OR, utilizando o processador IBM-VIGO com entrada $ 01\rangle$	68
Figura 59 – Resultado para a porta OR simulado em um processador quântico teórico com entrada $ 01\rangle$	68
Figura 60 – Arquitetura IBM-OURENSE	69
Figura 61 – Resultado para a porta XOR, utilizando o processador IBM-OURENSE com entrada $ 10\rangle$	69
Figura 62 – Resultado para a porta XOR simulado em um processador quântico teórico com entrada $ 10\rangle$	70
Figura 63 – Arquitetura IBM-qx2	70
Figura 64 – Resultado para a porta OR, utilizando o processador IBM-qx2 com entrada $ 00\rangle$	71
Figura 65 – Resultado para a porta OR simulado em um processador quântico teórico com entrada $ 00\rangle$	71
Figura 66 – Medição do estado antes da aplicação da primeira CNOT. Observa-se que para 8192 <i>trials</i> , em aproximadamente 90% destes, o resultado seria o correto, $ 00110\rangle$	73
Figura 67 – Teste de erro na primeira utilização da porta CNOT no circuito do QPerceptron.	74
Figura 68 – Teste de erro na primeira utilização da porta CNOT no circuito do QPerceptron, com uma configuração diferente de qubits.	75
Figura 69 – Teste de erro na segunda utilização da porta CNOT no circuito do QPerceptron.	76
Figura 70 – Teste de erro na segunda utilização da porta CNOT no circuito do QPerceptron, com uma configuração diferente de qubits.	77
Figura 71 – Teste de erro na terceira utilização da porta CNOT no circuito do QPerceptron.	78

Lista de tabelas

Tabela 1 – Notações algébricas.	19
Tabela 2 – tabela verdade OR	49
Tabela 3 – tabela verdade AND	55
Tabela 4 – tabela verdade XOR	58

Sumário

1	Introdução	12
2	Fundamentação Teórica	14
2.1	Noções de computação quântica	14
2.1.1	Tipos de computadores quânticos	17
2.1.2	Poder computacional quântico	18
2.2	Noções de mecânica quântica	19
2.2.1	Portas Quânticas	23
2.2.1.1	Portas de 1 Qubit	24
2.2.1.2	Portas de 2 ou mais qubits	26
2.3	Aprendizagem de Máquina	29
2.4	Redes Neurais	31
2.4.1	Perceptron	31
2.5	Computação Quântica e Aprendizagem de máquina	36
2.5.1	Aprendizagem de máquina quântica	36
2.5.2	Limitação do Perceptron clássico	38
3	Desenvolvimento, resultados e discussões	41
3.1	Desenvolvimento	41
3.1.1	Algoritmo de aprendizagem do Perceptron Quântico	41
3.2	Experimento	48
3.2.1	Experimento 1:	48
3.2.2	Experimento 2:	64
3.2.3	Decoerência nos processadores quânticos da IBM Q	72
4	Conclusão	79
A	Apêndice	80
A.1	Algoritmos quânticos	80
A.1.1	Algoritmo de Groover	80
A.1.2	Algoritmo de transformada de Fourier quântica de Shor	81
	Referências	82

1 Introdução

“Existe uma teoria que diz que, se um dia alguém descobrir exatamente para que serve o Universo e por que ele está aqui, ele desaparecerá instantaneamente e será substituído por algo ainda mais estranho e inexplicável. Existe uma segunda teoria que diz que isso já aconteceu.” (ADAMS et al., 2004)

Desde a mecânica dos corpos celestes até a mecânica de partículas subatômicas, as teorias físicas tentaram entender como o universo se comporta, o que tem contribuído significativamente, para as revoluções tecnológicas.

No final do século XVIII, com a Primeira Revolução industrial, as tecnologias das máquinas mecânicas e a vapor se basearam nas Leis de Newton, Bernouli entre outros cientistas, impactando grandes transformações no mundo. Em meados do século XIX, as novas teorias físicas que ganharam importância foram a elétrica e a magnética, cujos precursores foram Faraday, Tesla e Maxwell, que contribuíram, fortemente, na segunda revolução industrial para o desenvolvimento das máquinas elétricas e eletromagnéticas.

Na metade do século XX, a academia científica acreditava que as teorias físicas, em breve, seriam capazes de fornecer uma compreensão completa do mundo. Contudo, esse paradigma foi quebrado, devido ao surgimento de uma nova teoria, denominada quântica, que foi capaz de explicar o entendimento da natureza atômica e se tornou uma das responsáveis pela terceira revolução industrial.

A teoria quântica não é de simples compreensão, pois entidades quânticas não possuem propriedades objetivamente definidas: suas propriedades só podem ser especificadas e fundamentadas, por meio da matemática probabilística. Porém prevê alguns resultados de experimentos e fenômenos naturais com uma precisão insuperável por qualquer outra teoria da ciência atual.

Paralelo ao surgimento da teoria quântica, outra teoria também surgiu no século XX, a teoria da informação, tendo como precursor Shannon. Esta considera a comunicação como um problema matemático fundamentado nas teorias de probabilidade. Com o decorrer dos estudos, os físicos perceberam que havia uma interpretação alternativa que combinava a teoria da informação com a teoria quântica. A união da teoria da informação com a teoria quântica possibilitou dois novos ramos de estudos: a computação quântica e a informação quântica.

Ao mesmo tempo que a teoria quântica e a teoria da informação ganhavam força, uma nova área da ciência despontou em 1959, com Arthur Samuel, professor do MIT, que buscou formas de criar uma máquina capaz de aprender sem ter a necessidade de ser

programada explicitamente. A denominada “aprendizagem de máquina” nada mais é do que algoritmos capazes de realizar previsões futuras e de aprender com seus erros, com objetivo de se auto ajustar para melhorar as previsões futuras gradativamente. Dentro desta nova área, vários algoritmos foram sendo criados, baseados nesta, um deles foi o Perceptron, em 1957, desenvolvido pelo cientista Frank Rosenblatt.

O Perceptron era capaz de ser treinado para classificar certos tipos de padrões de acordo com os dados de entrada. Ele conseguia fazer isso por meio da divisão do espaço de entrada em regiões distintas para cada uma das classes existentes. Durante o treinamento o Perceptron aprendia a partir dos erros calculados na saída, adequando-se à teoria da aprendizagem de máquina de Arthur Samuel.

Entretanto, o Perceptron não era capaz de solucionar problemas mais complexos. A fim de superar essa limitação empregou-se uma rede interligada de Perceptrons, intitulada redes neurais. As redes neurais são custosas no que diz respeito a performance computacional, ou seja, quanto maior for o número de dados de entrada maior será o custo.

Portanto, a teoria de Rosenblatt possui, ainda, limitações de aplicação. Uma possível solução seria reformular o algoritmo do Perceptron para uma nova versão que possa superar a primeira. O presente trabalho busca demonstrar uma aplicação da computação quântica, demonstrando seu poder computacional em comparação com a computação atual, abordando uma re-formulação do algoritmo clássico do Perceptron para uma nova versão que é embasada nos princípios da computação quântica.

No primeiro capítulo deste trabalho, será apresentada uma breve fundamentação teórica da computação quântica, mecânica quântica, aprendizagem de máquina e redes neurais; Em seguida será desenvolvido uma reformulação do algoritmo do Perceptron clássico para uma nova versão quântica e será realizado dois experimentos, um no processador quântico teórico e outro no processador quântico da IBM-Q. Por fim será apresentado o resultado obtido nos experimentos e eventuais erros que foram gerados pelo processador da IBM-Q.

2 Fundamentação Teórica

2.1 Noções de computação quântica

A busca de avanços tecnológicos na área da computação é fato, diante da crescente quantidade de produção dados que gera demanda tecnológica para processá-los. Ao mesmo tempo, há progressiva demanda por performance nos Circuitos Integrados (CI) de computadores. Segundo a Lei de Gordon Moore, o número de transistores seria dobrado a cada ano em um circuito integrado de uma mesma área. A “miniaturização” dos componentes eletrônicos, contudo, gera duas consequências:

a) Elétrons movendo-se em circuitos pequenos (CIs) geram um maior aquecimento nestes circuitos ([JAEGER, 2018](#)).

b) Estruturas eletrônicas usadas nos chips, estão com tamanho inferior a 10 nm, isso corresponde a aproximadamente 40 átomos. Em transistores deste tamanho, as leis da mecânica quântica prevalecem e acabam por provocar comportamentos não esperados na circuitaria ([JAEGER, 2018](#)).

Em face das consequências apresentadas em razão da “miniaturização”, a partir do roteiro de 2016, apresentado pela indústria de semicondutores para os anos seguintes, o desenvolvimento de CIs não seguiria mais os fundamentos da lei de Moore.

“Abandonar a lei de Moore será talvez a melhor coisa que pode acontecer para a evolução dos computadores, porque deverá acelerar o distanciamento face a uma arquitetura envelhecida”. Essa é a visão do proeminente cientista Richard Stanley Williams. Stanley refere-se ao cenário da computação atual como uma “arquitetura envelhecida”, pois os princípios e funcionalidades dos computadores atuais estão ainda embasados nos conceitos da física clássica, descritos por Alan Turing e Jonh von Neumann ([ADAMI; DOWLING, 2002](#)).

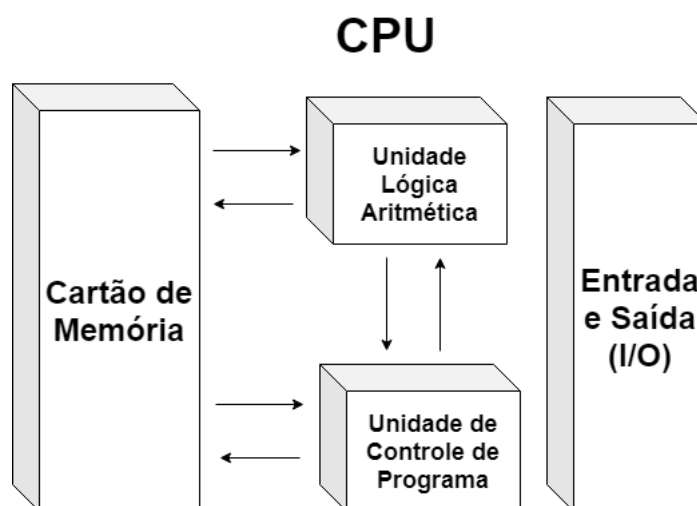
Os primeiros computadores da década de 1940 eram formados por três elementos: tubos, capacitores e transistores, sendo o último, peça chave para qualquer computador da atualidade. Contudo, ainda hoje, os transistores permanecem com a mesma aplicação, apesar de terem reduzido de tamanho e melhorado a sua velocidade de resposta.

Por sua vez a arquitetura básica dos computadores permanece fundamentada na arquitetura de Von Neumann, Figura 1, em que o centro do computador era o cartão de memória, que continha tanto o programa quanto os dados a serem processados sequencialmente.

O processamento de dados digitais nos computadores consiste em converter a

informação na forma de bits em uma saída na forma de arranjos de bits, estes corresponde à unidade básica de informação que o computador processa (JAEGER, 2018). Os componentes eletrônicos básicos para processar os bits são conhecidos como portas lógicas, que são construídos por componentes físicos como transístores. Tais portas lógicas são conectadas entre si de diferentes maneiras para resultar em operações mais complexas, por exemplo, a adição e multiplicação de bits (VAHID, 2009; JAEGER, 2018).

Figura 1 – Arquitetura de Von Neumann



Entretanto, a simplicidade de como a informação é processada tem um preço: um grande número de bits são solicitados, para lidar com a quantidade de dados necessários para as aplicações complexas realizadas nos computadores. A fim de realizar essas aplicações, diversos transístores são precisos para construir portas lógicas, hoje, por exemplo, os CIs operam com bilhões de transístores. Para criar estes CIs é indispensável empacotar os transístores em dimensões microscópicas, sendo estes construídos em camadas de material semiconductor. Neste momento conceitos práticos de física quântica entram em ação para explicar o funcionamento dos transístores (JAEGER, 2018; RIEFFEL; POLAK, 2011).

Porém, o entendimento sobre física quântica não era necessário para a revolução digital dos transístores, mas um pré-requisito para a extrema "minituarização" dos CIs (JAEGER, 2018). Portanto, no caso dos computadores atuais, os efeitos quânticos desempenham um papel significativo, o que não altera o fato de que sua funcionalidade seja baseada inteiramente nos princípios da física clássica, pois a influência destes efeitos permanece confinada ao domínio da implementação de baixo nível. Estes efeitos não tiveram nenhum desempenho sobre como a computação ou a comunicação foi pensada ou estudada (RIEFFEL; POLAK, 2011).

No século XX, o cientista Richard Feynman, durante uma de suas palestras, fez o questionamento, se o mundo quântico poderia ser simulado por meio de um computador convencional, (*Simulating Physics with Computer*) (FEYNMAN, 1982). O problema de

simular o mundo quântico são as probabilidades associadas aos estados quânticos, uma vez que as variáveis quânticas não assumem valores fixos. De fato, a qualquer momento, as variáveis quânticas preenchem todo um espaço matemático de possíveis estados. Isso aumenta o escopo dos cálculos exponencialmente (JAEGER, 2018).

Feynman, reconhecendo que um computador convencional não consegue simular o mundo quântico, apontou a necessidade de um computador cujos estados internos são variáveis quânticas, ou seja, um computador que explora as propriedades quânticas e apresenta uma estrutura e funcionalidade substancialmente diferente da arquitetura dos computadores atuais. Dessa forma, em vez de processar, sequencialmente, bit por bit como uma máquina de Turing, o computador quântico calcularia em paralelo os inúmeros estados assumidos simultaneamente pelas variáveis quânticas. Nas palavras de Feynman: *“O processamento e armazenamento de informações não ocorrerão mais por meio de correntes de elétrons, mas serão baseados no controle e na direção de partículas quânticas únicas”* *Simulating Physics with Computer*) (FEYNMAN, 1982).

O computador quântico une todo o arcabouço teórico da Informação quântica, mecânica quântica e teoria da computação desenvolvidos no século XX. A computação quântica não visa mudar o aparato ou “substrato” físico, mas transforma o entendimento da computação em si.

A mudança do computador atual para o computador quântico inicia no nível mais básico: a unidade fundamental do computador quântico não é mais o bit, mas o bit quântico ou qubit, marcando o estado da arte da computação quântica (NIELSEN; CHUANG, 2002).

Um bit, em termos de sistemas físicos, pode ser relacionado como: um capacitor carregado, um estado do transistor e uma polarização magnética. Enquanto um qubit é tratado em sistemas físicos como: spin do elétron, estado atômico e estados de circuitos supercondutores. Qubits não são somente 0 ou 1 como os bits, estes podem aceitar ambos os estados simultaneamente, bem como todos os valores entre eles. Logo, o qubit contém mais informações e a diferença entre o bit e qubit pode ser vista na Figura 2. Essa particular capacidade dos qubits acontece devido as duas propriedades que são possíveis apenas no mundo da mecânica quântica: Superposição e Emaranhamento.

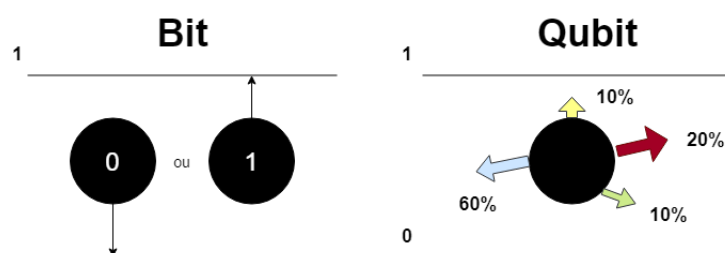
a) Superposição: os estados quânticos podem existir em superposições de estados classicamente exclusivos. Isso permite que um qubit assumam o estado 0 e 1 simultaneamente (JAEGER, 2018).

b) Emaranhamento: vários qubits podem ser trazidos para estados emaranhados, os quais estão acoplados por uma mola invisível e interligados entre si, - “ação assustadora à distância” termo empregado ironicamente por Einstein a fim de expressar sua descrença em relação a esse fenômeno quântico. Os qubits emaranhados estão em algum tipo de contato

direto um com o outro, mesmo quando estão espacialmente bem separados (JAEGER, 2018).

Na computação clássica, apenas resultados especificamente definidos estão disponíveis, limitados pelo design de um algoritmo. Respostas quânticas, que estão em quantidades chamadas amplitudes, são probabilísticas, significando que por causa da superposição e emaranhamento, múltiplas respostas podem ser consideradas em um dado cálculo (FURSMAN, 2007).

Figura 2 – Diferença entre Bit e Qubit



2.1.1 Tipos de computadores quânticos

Existem três tipos de computadores quânticos de acordo com Carl De Torres, pesquisador da IBM:

a) *The quantum annealer*: é a forma menos poderosa, mais restritiva de computador quântico e mais fácil de construir. Contudo, só é possível executar uma função específica. O *quantum annealer* foi desenvolvido com sucesso pela empresa canadense D-Wave, Figura 3. Em dezembro de 2015, o Google deu credibilidade a esse tipo de computador, quando revelou testes mostrando que o computador quântico D-Wave era cerca de 3.600 vezes mais rápido que um supercomputador na solução de problemas específicos e complexos¹.

b) *The analog quantum*: capaz de simular interações quânticas complexas que são intratáveis para qualquer máquina convencional conhecida. Supõe-se que o computador quântico analógico contenha algo entre 50 e 100 qubits.

c) *The universal quantum*: é o mais poderoso, geral e difícil de construir, apresentando vários desafios técnicos. As estimativas atuais indicam que esta máquina compreenderá mais de 100.000 qubits físicos.

Além dos tipos de computadores quânticos descritos grupos de pesquisa como Google, IBM dentro outros procuram realizar computadores quânticos injetando elétrons em laços de supercondutores circulares, os chamados dispositivos de interferência quântica supercondutora ou SQUIDs, sendo estes interrompidos por camadas muito finas de isolante (JAEGER, 2018).

¹ Fonte: <https://www.visualcapitalist.com/three-types-quantum-computers/>. Acesso: Julho 2019

Figura 3 – Computador Quântico Dwave



Fonte: <https://techcrunch.com/2017/01/24/d-wave-ups-its-quantum-annealing-game-to-2000-qubits/>
(Acesso: 25/09/2019)

2.1.2 Poder computacional quântico

O emaranhamento de muitos qubits combinados em registros quânticos, permite a paralelização de alto nível das operações que tornam os computadores quânticos tão poderosos. Diferentemente do computador convencional, cujo o seu poder computacional aumenta linearmente com o número de componentes eletrônicos, o poder computacional de um computador quântico aumenta exponencialmente com o número de qubits usados.

Além disso, algumas limitações encontradas nos computadores atuais podem ser resolvidos pelos computadores quânticos. Por exemplo:

a) Criptografia: quase todos os métodos de criptografia comuns são baseados em fatorar o produto de dois primos muito grandes. Para descriptografar a mensagem, é necessário determinar que dado dois números primos, um determinado número é composto. Todavia, além de depender do tamanho do número, essa tarefa ainda poderá não ser resolvida por um computador clássico. Em 1994, Peter Shor desenvolveu um algoritmo para uso com um computador quântico que era capaz de fatorar os produtos de números primos muito grandes em seus divisores em questão de minutos (JAEGER, 2018).

b) Aplicações no campo da Inteligencia Artificial: as redes neurais profundamente empregadas nesse campo têm problemas de otimização combinatória que o computador quântico pode solucionar melhor e mais rápido que qualquer computador clássico (JAEGER, 2018).

c) Busca em grandes bancos de dados: Ao pesquisar conjuntos de dados não classificados, um computador clássico é forçado a examinar cada ponto de dados individualmente. O tempo de pesquisa, portanto, aumenta linearmente com o número de pontos de dados. Para grande demanda de dados, o número de etapas de computação necessárias para realizar esta tarefa é muito grande para um computador clássico. No ano de 1996, o cientista da computação Lov Grover publicou um algoritmo quântico, para o qual o número de etapas computacionais necessárias aumenta apenas como a raiz quadrada do número de

pontos de dados. Ou seja, Em vez de demorar mil vezes mais para um bilhão de entradas de dados no computador convencional, a mesma tarefa poderia ser executada cerca de 30 vezes mais rápida ao aplicar-se um computador quântico com o algoritmo de Grover (JAEGER, 2018).

2.2 Noções de mecânica quântica

Tabela 1 – Notações algébricas.

Notção	Descrição
$ \psi\rangle$	Vetor ou ket
$\langle\psi $	Vetor dual ou bra
$\langle\varphi \psi\rangle$	Produto interno
$ \psi\rangle \otimes \varphi\rangle$	Produto tensorial
$A^\dagger = (A^T)^*$	Conjugado hermitiano

A mecânica quântica não é tão difícil de aprender, apesar de ser tratada como tal. A sua reputação decorre da dificuldade de compreender alguns fenômenos que não se aplicam nas experiências cotidianas. Além disso, a mecânica quântica é uma teoria, no sentido matemático, governada por um conjunto de postulados. As consequências dos postulados descrevem o comportamento dos sistemas quânticos, que levando aos vários aparentes paradoxos, como o paradoxo EPR (RIEFFEL; POLAK, 2000) (NIELSEN; CHUANG, 2002)(RIEFFEL; POLAK, 2011).

Conhecimentos gerais em álgebra linear são necessários para compreensão da mecânica quântica, como as apresentadas na Tabela 1 (NIELSEN; CHUANG, 2002).

Os objetos básicos da álgebra linear são espaços vetoriais e as suas operações lineares. O espaço de estados de um sistema quântico, consiste nas: posições, momentos, polarizações, rotações dentre outros, sendo o sistema modelado por um espaço de Hilbert (\mathcal{H}) de funções de onda. Essas funções de onda vão além do escopo deste trabalho, visto que para a computação quântica se precisa lidar somente com sistemas quânticos finitos e espaços vetoriais dimensionais finitos.

A notação de álgebra linear utilizada na computação quântica é a notação Dirac, criada por Paul Dirac. Em matemática e física, comumente, os vetores são diferenciados dos escalares por colocar uma seta acima do símbolo de identificação: \vec{c} ou então escritos em negrito: **c** (KAYE et al., 2007).

Como \mathcal{H} é de dimensão finita, pode-se escolher uma base e, alternativamente, representar vetores (kets) nessa base como vetores de colunas finitas e representar os operadores com matrizes finitas, como será descrito nas próximas subseções. Foi escolhido

fixar uma base conveniente e referir-se a ela como a base computacional, é representada pela Equação 2.1:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.1)$$

Postulado 1: Associado a qualquer sistema físico isolado, há um espaço vetorial complexo com produto interno (ou seja, um espaço de Hilbert) conhecido como espaço de estados do sistema. O sistema é completamente descrito por seu vetor de estado, que é uma unidade do vetor no espaço de estado do sistema.

O sistema mecânico quântico mais simples e importante para a computação quântica sendo a unidade fundamental de informação no processamento de informações quânticas é o qubit. O qubit pode ser escrito como: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ onde α, β são números complexos e $|0\rangle$ é um vetor $[0 \ 1]^T$ e $|1\rangle$ é um vetor $[1 \ 0]^T$. Pode-se escrever o qubit como um vetor unitário no espaço dimensional \mathcal{C}^2 de acordo com a Equação 2.2:

$$\begin{aligned} |\psi\rangle &= \alpha|0\rangle + \beta|1\rangle \\ |\psi\rangle &= \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ |\psi\rangle &= \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \end{aligned} \quad (2.2)$$

ou considerando que $\alpha = a + ib$ e $\beta = c + id$, a Equação 2.3 representa essa nova forma de reescrever a Equação 2.2:

$$|\psi\rangle = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} i \\ 0 \end{bmatrix} + c \begin{bmatrix} 0 \\ 1 \end{bmatrix} + d \begin{bmatrix} 0 \\ i \end{bmatrix} \quad (2.3)$$

A interpretação física de um qubit é que ele está simultaneamente nos estados $|0\rangle$ e $|1\rangle$ (Superposição). Isso faz com que a quantidade de informação armazenada no estado $|\psi\rangle$ possa ser infinita, entretanto essa quantidade infinita de informação está no nível quântico (MOTTA; CARVALHO; MACULAN, 2005). Porém o processo de medida colapsa o estado de um qubit, fazendo-o assumir o estado de $|0\rangle$ com probabilidade $|\alpha|^2$, ou o estado $|1\rangle$, com probabilidade $|\beta|^2$, ou seja a condição para que $|\psi\rangle$ seja um vetor unitário é:

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

O estado $|\psi\rangle$ de um sistema mecânico muda com o tempo através da descrição dada pelo postulado 2;

Postulado 2: A evolução de um sistema quântico fechado é descrita por uma transformação unitária. Isto é, o estado $|\psi\rangle$ do sistema no tempo t_1 está relacionado com o estado $|\psi'\rangle$ do sistema em um tempo t_2 por uma operação unitária U que depende unicamente dos tempos t_1 e t_2 .

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

A evolução dos sistemas que não interagem com o resto do mundo está muito bem definida, mas também deve haver momentos em que o experimentalista utiliza seu equipamento experimental e observa o sistema para descobrir o que está acontecendo com este, ou seja, uma interação que não deixa mais o sistema fechado e portanto, o sistema não está mais sujeito à evolução unitária. Para explicar o que acontece quando observa-se o sistema, o Postulado 3 fornece um meio para descrever os efeitos de medições em sistemas quânticos (NIELSEN; CHUANG, 2002).

Postulado 3: As medidas quânticas são descritas por uma coleção M_m de operadores de medição. Esses são operadores que atuam no espaço de estados do sistema medido. O índice m se refere aos resultados da medição que pode ocorrer no experimento. Se o estado do sistema quântico for $|\psi\rangle$ imediatamente antes da medição, a probabilidade de o resultado m ocorrer é dado por:

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

e o estado do sistema após a medida é:

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

Até agora um qubit foi descrito como $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, onde $|0\rangle$ e $|1\rangle$ são descritos na base natural de dimensão 2 do espaço complexo de Hilbert, \mathcal{H} . Os vetores de base desempenham um papel poderoso no estudo de transformações lineares, a definição de base significa que é possível expressar qualquer vetor como uma combinação linear de vetores bases com os respectivos escalares (LOCEFF, 2015), como mostra a Equação 2.4:

$$|\psi\rangle = \sum_{k=1}^N \beta_k |b_k\rangle \quad (2.4)$$

Se for aplicado uma transformação U em $|\psi\rangle$, obtém-se:

$$U |\psi\rangle = U \left(\sum_{k=1}^N \beta_k |b_k\rangle \right) = \sum_{k=1}^N \beta_k U(|b_k\rangle)$$

Visto até agora os qubits com mais detalhes, existe uma particularidade que pode ser utilizada no princípio da medição, Conforme o estado:

$$|\psi\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{e^{i\theta}}{\sqrt{2}} |1\rangle$$

Se fosse medido esse qubit na base natural computacional, o resultado seria 0 com probabilidade $1/2$ e 1 com probabilidade $1/2$. Essa medida diz apenas sobre as normas das amplitudes dos estados bases, porém existe uma maneira de obter a probabilidade de 0 ou 1 pela informação da fase θ . Para isso é preciso mudar a base na qual $|\psi\rangle$ está descrito, para uma nova base, esta definida pela Equação 2.2:

$$\begin{aligned} |+\rangle &\equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &\equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \quad (2.5)$$

Reescrevendo $|0\rangle$ e $|1\rangle$ em termos da nova base, como mostra a Equação 2.2:

$$\begin{aligned} |0\rangle &= \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \\ |1\rangle &= \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) \end{aligned} \quad (2.6)$$

Substituindo a Equação 2.2 em $|\psi\rangle$, obtém-se:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) + \frac{e^{i\theta}}{\sqrt{2}}(|+\rangle - |-\rangle) \\ |\psi\rangle &= \frac{1+e^{i\theta}}{\sqrt{2}} |+\rangle + \frac{1-e^{i\theta}}{\sqrt{2}} |-\rangle \end{aligned}$$

Considerando a relação de Euler $e^{i\theta} = \cos(\theta) + i\sin(\theta)$, conclui-se que a probabilidade de obter o estado $|+\rangle$ é de $\cos^2(\theta/2)$ e a probabilidade de obter o estado $|-\rangle$ é de $\sin^2(\theta/2)$. A medição na base $(|-\rangle, |+\rangle)$, portanto, permite que a probabilidade de medição do qubit seja baseada na sua fase θ .

Além dos postulados descritos, existe uma importante propriedade para descrever o funcionamento das operações quânticas nos qubits, essa é a transformação unitária.

Transformação Unitária , Transformação quântica ou operação quântica, visa à mapear do espaço de estado de um sistema quântico para ele mesmo, detalhadamente considera-se como uma transformação linear em \mathcal{H} uma transformação que mapeia um vetor

normalizado para outro vetor normalizado. Medições não são transformações quânticas neste sentido. A linearidade significa para uma transformação quântica U :

$$U(a_1 |\psi_1\rangle + \cdots + a_k |\psi_k\rangle) = a_1 U |\psi_1\rangle + \cdots + a_k U |\psi_k\rangle$$

Vetores de tamanhos unitários devem ser mapeados para vetores de tamanhos unitários, o que implica que um sub-espço ortogonal mapeia-se para um sub-espço ortogonal. Para qualquer transformação U , seu adjunto U^\dagger deve ser igual a sua inversa, precisamente $U^\dagger = U^{-1}$, para uma transformação ser linear, a unitariedade de U significa que: $\langle x | U^\dagger U | y \rangle = \delta_{xy}$.

Geometricamente, todas as transformações de estados quânticos são rotações do espaço vetorial complexo associado com o espaço de estado quântico. Desde que U^\dagger seja unitário se e somente se U for, pode-se afirmar que U é unitário se e somente se suas linhas forem ortonormais. O produto $U_1 U_2$ de duas transformações unitárias é unitária, o produto tensorial $U_1 \otimes U_2$ é uma transformação unitária do espaço $X_1 \otimes X_2$ se U_1 e U_2 forem transformações unitárias de X_1 e X_2 respectivamente, porém combinações lineares de operadores unitários geralmente não são unitárias.

Uma consequência da condição unitária é que toda transformação no estado quântico é reversível. Nos modelos de circuitos padrões de computação quântica, toda computação é realizada por transformações quânticas, com medições sendo feitas apenas no final para leitura dos resultados. As portas quânticas que serão descritas na próxima secção são transformações unitárias aplicadas a um ou mais qubits.

2.2.1 Portas Quânticas

Portas quânticas são transformações unitárias do estado quântico, por sua vez, a sequência de portas são denominadas de circuitos quânticos. Assim, como qualquer cálculo em um computador clássico pode ser dividido em uma sequência de portas lógicas que atuam apenas em alguns bits clássicos de cada vez, também qualquer cálculo quântico pode ser dividido em uma sequência de portas quânticas que atuam apenas em alguns qubits por vez. A principal diferença é que, enquanto as portas lógicas clássicas manipulam os valores de bits clássicos, 0 ou 1, as portas quânticas podem manipular estados quânticos em superposições arbitrárias, que frequentemente também são emaranhados. Assim, as portas quânticas são, consideravelmente, mais variadas do que as portas lógicas da computação clássica ([WILLIAMS, 2010](#)).

As propriedades essenciais das portas quânticas fluem imediatamente a partir do fato de que elas são descritas por matrizes unitárias. Uma matriz, U , é unitária se, e somente se, a sua inversa é igual a sua transposição do conjugado: $U^{-1} = U^\dagger$. Para qualquer porta quântica U , $U^\dagger U = I$ garante que sempre é possível desfazer uma porta quântica,

isto é, que uma porta quântica é logicamente reversível (WILLIAMS, 2010)(BENENTI; CASATI; STRINI, 2004).

A seguir, serão abordados portas quânticas que atuam em um único q-bit, e portas quânticas em estados quânticos de mais de um qubit, ou seja, estado emaranhado.

2.2.1.1 Portas de 1 Qubit

a) Porta quântica NOT ou X: A porta quântica NOT ou matriz Pauli X é sinônimo da porta NOT clássica (reversível), esta atua trocando as amplitudes de um estado quântico.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Aplicando a porta quântica NOT em $|\psi\rangle = [\alpha, \beta]^T$:

$$X|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Pode-se representar a porta quântica NOT através da Equação: $X|0\rangle = |1\rangle$ e $X|1\rangle = |0\rangle$. O operador X muitas vezes é chamado de bit-flip, porque troca os coeficientes das bases de estados computacionais, $\alpha \leftrightarrow \beta$. O símbolo da porta X é representado pelo símbolo da Figura 4:

Figura 4 – Porta quântica NOT e seu funcionamento em um estado $|\psi\rangle$

$$\begin{array}{c} \text{---} \boxed{X} \text{---} \\ \beta|1\rangle + \alpha|0\rangle \text{---} \boxed{X} \text{---} \beta|0\rangle + \alpha|1\rangle \end{array}$$

b) Porta quântica Z: A porta quântica Z é definida como:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

A porta Z nega a segunda amplitude do vetor de estado quântico, deixando a primeira sem alterações. Aplicando em $|\psi\rangle = [\alpha, \beta]^T$:

$$Z|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix}$$

A porta quântica Z é chamada de "phase flip", pois muda a fase relativa das duas amplitudes do vetor $|\psi\rangle$, já que $-1 = e^{i\pi}$, então multiplicando β por $\pi = 180^\circ$, rotaciona β no plano complexo. O símbolo da porta Z é representado pela Figura 5:

Figura 5 – Porta quântica Z e seu funcionamento em um estado $|\psi\rangle$

$$\begin{array}{c} \text{---} \boxed{Z} \text{---} \\ \beta |1\rangle + \alpha |0\rangle \text{---} \boxed{Z} \text{---} \beta |1\rangle - \alpha |0\rangle \end{array}$$

c) Porta quântica Y: A porta quântica Y também chamada "bit-and-phase flip", troca ambos os bits e suas fases relativas, sendo definida como:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Aplicando em $|\psi\rangle = (\alpha, \beta)^T$:

$$Y |\psi\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = -i \begin{bmatrix} \beta \\ -\alpha \end{bmatrix}$$

O símbolo da porta Y é representado pela Figura 6:

Figura 6 – Porta quântica Y e seu funcionamento em um estado $|\psi\rangle$

$$\begin{array}{c} \text{---} \boxed{Y} \text{---} \\ \beta |1\rangle + \alpha |0\rangle \text{---} \boxed{Y} \text{---} -i\beta |0\rangle + i\alpha |1\rangle \end{array}$$

d) Porta quântica H: A porta quântica H é definida como:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Aplicando na base computacional $|0\rangle$ e $|1\rangle$:

$$H |0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$H |1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Pode-se notar que a porta Hadamard "rotaciona" os kets da base-z para kets na base-x. Aplicando para $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$:

$$H |\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha + \beta \\ \alpha - \beta \end{pmatrix}$$

Figura 7 – Porta quântica H e seu funcionamento em um estado $|\psi\rangle$

$$\begin{array}{c} \text{---} \boxed{H} \text{---} \\ \beta |1\rangle + \alpha |0\rangle \text{---} \boxed{H} \text{---} \left(\frac{\alpha+\beta}{\sqrt{2}} \right) |0\rangle - \left(\frac{\alpha-\beta}{\sqrt{2}} \right) |1\rangle \end{array}$$

O símbolo da porta H é representado pela Figura 7:

Uma das portas mais úteis da computação quântica é a Porta Hadamard, H, pois permite, carregar exponencialmente muitos índices em um computador quântico usando apenas operações linearmente numerosas (WILLIAMS, 2010).

A utilidade do porta Hadamard deriva do fato de que, ao aplicar, paralelamente, uma porta Hadamard separada para cada um dos n-qubits, cada um inicialmente no estado $|0\rangle$, pode-se criar uma superposição de n-qubit contendo 2^n termos de base computacional. Estes representam todas as cadeias de bits possíveis que se pode escrever usando n qubits.

2.2.1.2 Portas de 2 ou mais qubits

Um sistema de dois qubits se refere ao produto dos espaços $\mathcal{H} \otimes \mathcal{H}$. Para um qubit foi estabelecido que a base computacional seria $|0\rangle$ e $|1\rangle$, para dois qubits os estados da base computacional serão:

$$\begin{aligned} |0\rangle \otimes |0\rangle &\leftrightarrow |00\rangle \\ |0\rangle \otimes |1\rangle &\leftrightarrow |01\rangle \\ |1\rangle \otimes |0\rangle &\leftrightarrow |10\rangle \\ |1\rangle \otimes |1\rangle &\leftrightarrow |11\rangle \end{aligned}$$

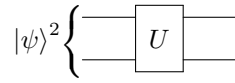
Um operador binário quântico é uma transformação unitária U, no sistema de dois qubits $\mathcal{H} \otimes \mathcal{H}$. Toda porta binária tem duas linhas, uma para cada entrada de qubit e duas linhas de saída. Um estado de dois qubits definido na base computacional é representado como uma superposição de quatro vetores bases de $\mathcal{H} \otimes \mathcal{H}$:

$$|\psi\rangle^2 = \alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle$$

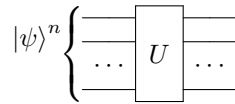
onde:

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

A transformação unitária para dois qubits é realizada da mesma forma para um qubit, porém, agora U está em um sistema $\mathcal{H} \otimes \mathcal{H}$ como mostra a Figura 8.

Figura 8 – Aplicação de uma porta quântica U em dois qubits

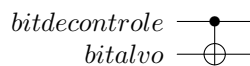
Para sistemas maiores que 2 qubits, a definição da transformação unitária U é definida como $\mathcal{H} \otimes \dots \otimes \mathcal{H}$ ou \mathcal{H}^n , onde n é a quantidade de qubits do sistema, como mostra a Figura 9.

Figura 9 – Aplicação de uma porta quântica U em n qubits

Para executar cálculos não triviais, geralmente é necessário alterar a operação aplicada a um conjunto de qubits, dependendo dos valores de outro conjunto de qubits. As portas que implementam essas operações do tipo "se-então-outro" são chamados de portas controladas. Alguns exemplos de portas controladas são CNOT (NÃO controlado), FREDKIN (SWAP controlado) e TOFFOLI (controlado-controlado-NÃO). A justificativa para chamar esses portões de "controlados" decorre de seus efeitos nos estados da base computacional. Sendo as portas FREDKIN e TOFFOLI aplicadas a um sistema de 3 qubits.

a) CNOT: Transforma os estados da base computacional de forma que o segundo qubit seja negado se, e somente se, o primeiro qubit estiver no estado $|1\rangle$. Portanto, o valor do segundo qubit (chamado qubit “alvo”) é controlado pelo primeiro qubit (chamado qubit de “controle”) (WILLIAMS, 2010), como mostra a Figura 10.

Figura 10 – Porta quântica CNOT



Funcionamento da porta:

$$|00\rangle \longrightarrow |00\rangle$$

$$|01\rangle \longrightarrow |01\rangle$$

$$|10\rangle \longrightarrow |11\rangle$$

$$|11\rangle \longrightarrow |10\rangle$$

b) TOFFOLI: Também é possível ter portas controladas com múltiplos qubits de controle e múltiplos qubits de destino. A ação da porta TOFFOLI é negar o terceiro qubit (ou seja, o qubit alvo) se, e somente se, os dois primeiros qubits (os qubits de controle) estiverem no estado $|11\rangle$. Assim, a porta TOFFOLI possui dois qubits de controle e um qubit alvo ([WILLIAMS, 2010](#)), como mostra a Figura 11.

Figura 11 – Porta quântica TOFFOLI



Funcionamento da porta:

$$|000\rangle \longrightarrow |000\rangle$$

$$|001\rangle \longrightarrow |001\rangle$$

$$|010\rangle \longrightarrow |010\rangle$$

$$|011\rangle \longrightarrow |011\rangle$$

$$|100\rangle \longrightarrow |100\rangle$$

$$|101\rangle \longrightarrow |101\rangle$$

$$|110\rangle \longrightarrow |111\rangle$$

$$|111\rangle \longrightarrow |110\rangle$$

c) FREDKIN: Da mesma forma que na CNOT a ação da porta FREDKIN é similar à ação da CNOT, sendo que o segundo e o terceiro qubits são trocados, se, e somente se, o primeiro qubit estiver no estado $|1\rangle$. Portanto, executa uma operação denominada: SWAP controlada ([WILLIAMS, 2010](#)), como mostra a Figura 12.

Figura 12 – Porta quântica FREDKIN



Funcionamento da porta:

$$|000\rangle \longrightarrow |000\rangle$$

$$|001\rangle \longrightarrow |001\rangle$$

$$|010\rangle \longrightarrow |010\rangle$$

$$|011\rangle \longrightarrow |011\rangle$$

$$|100\rangle \longrightarrow |100\rangle$$

$$|101\rangle \longrightarrow |110\rangle$$

$$|110\rangle \longrightarrow |101\rangle$$

$$|111\rangle \longrightarrow |111\rangle$$

As noções de computação quântica e de mecânica quântica abordadas nos itens 2.1 e 2.2 são necessárias para subsidiar o tópico essencial deste trabalho, que é a aprendizagem de máquina quântica ou *Quantum Machine Learning*.

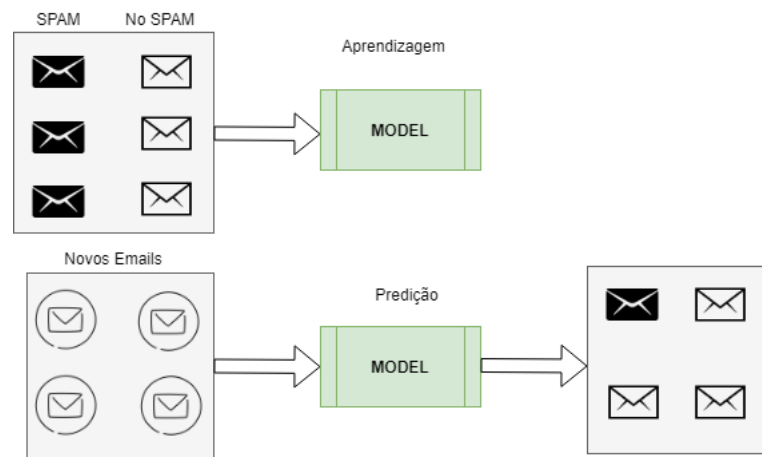
2.3 Aprendizagem de Máquina

A aprendizagem engloba grande leque de processos e, dessa forma, é difícil descrevê-la precisamente. Definições extraídas de dicionários incluem frases como: “ganhar conhecimento, ou entender algo através de estudo ou experiência”. Tratando-se de máquinas, pode-se dizer que aprendem quando mudam a estrutura, programa ou algoritmo de tal maneira que se espera que a performance melhore futuramente (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Algoritmos convencionais podem ser descritos na ciência da computação como uma sequência finita de ações executáveis que visam a obter uma solução para um determinado tipo de problema, onde o programador impõe regras e, a partir dos dados recebidos o algoritmo gera respostas. Entretanto para algumas tarefas um algoritmo convencional, por si só, não é capaz de solucioná-las. Por exemplo, dizer se um e-mail é um spam ou não, sabe-se que a entrada é um e-mail e a saída é uma resposta binária $S = [0,1]$ ($0 = \text{Não Spam}$ e $1 = \text{Spam}$), o problema é que nem todo e-mail tem o mesmo formato, que varia em relação tanto ao tempo quanto em atributos de tal forma que um algoritmo convencional não é capaz de resolver (PARSONS, 2005). Neste exemplo, são fornecidos dados prévios – e-mails que tenham ou não spam – com o objetivo de criar um modelo ou regra que aprenda com estes dados e seja capaz de diferenciar futuros e-mails, como mostra a Figura 13.

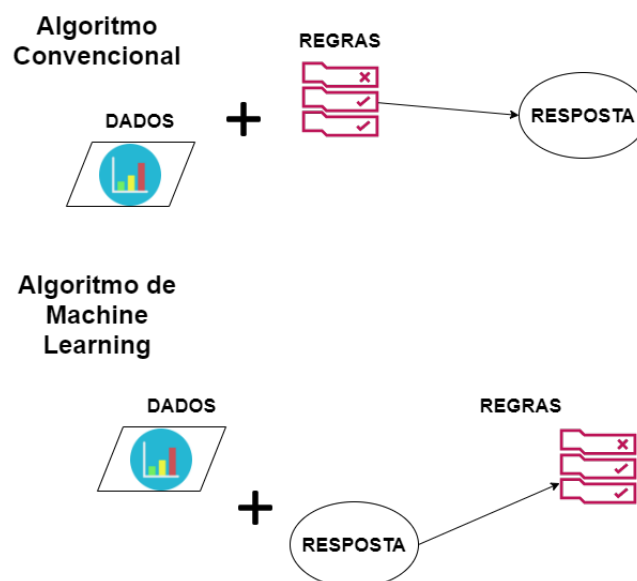
Sabe-se que não é possível fazer um modelo perfeito que consiga aprender o processo de classificar os novos e-mails sem cometer erros, mas acredita-se que ser viável construir uma aproximação boa e utilizável capaz de reconhecer certos padrões e regularidades que tornam o modelo satisfatório. Isto descreve o nicho da área de aprendizagem de máquina ou *Machine Learning*.

Figura 13 – Classificação de E-mails



Assim, pode-se dizer que *Machine Learning* é uma área interdisciplinar que usa fortemente teorias estatísticas, teoria da informação e otimizações para montar modelos matemáticos, visando à detecção automatizada de padrões significativos nos dados (SHALEV-SHWARTZ; BEN-DAVID, 2014).

O principal objetivo do *Machine Learning* é propor algoritmos otimizados que façam com que a máquina aprenda novas regras e se adapte a novas situações com o intuito de fazer inferências a partir de novos dados, ou seja, melhorar sua performance continuamente. A Figura 14 mostra a diferença entre um algoritmo convencional e um algoritmo de aprendizagem de máquina.

Figura 14 – Comparação entre os fluxogramas dos algoritmos convencionais e *Machine Learning*

O *Machine Learning* pode ser subdividido em tipos distintos de aprendizagem. Existe a aprendizagem supervisionada e não-supervisionada. Considera-se aprendizagem supervisionada quando se tem de antemão as variáveis de entrada e saída e utiliza-se o

algoritmo para que a máquina aprenda a função que mapeia a entrada para a saída. O objetivo é aproximar tão bem a função de mapeamento que, quando se tiver novos dados de entrada, o algoritmo irá prever a saída que mais se aproxima da resposta correta (tomando como base a função que mapeia a entrada na saída). Por sua vez, na aprendizagem não supervisionada se tem de antemão somente as variáveis de entrada, o objetivo então é modelar a estrutura ou distribuição subjacente nos dados para aprender mais sobre os mesmos. Diferentemente do aprendizado supervisionado, não há respostas corretas e não há uma função de mapeamento.

2.4 Redes Neurais

As redes neurais artificiais estão inseridas dentro da área de *Machine Learning*. Estes modelos computacionais foram desenvolvidos a partir de modelos conhecidos de sistemas nervosos biológicos, onde os neurônios artificiais são elementos computacionais inspirados nos modelos de neurônios biológico. A rede neural em geral possui capacidade de aquisição e manutenção do conhecimento (baseada em informações) e são formadas pela interligação de neurônios. Entre muitas vantagens desse modelo pode-se citar: a) adaptação por experiência: adaptação dos parâmetros internos da rede. b) tolerância a falhas: devido ao elevado nível de interconexões entre os neurônios artificiais ; c) facilidade de prototipagem: a implementação da maioria das arquiteturas neurais pode ser facilmente, dependendo da especificidade da aplicação, prototipada em hardware.

As redes neurais podem ser empregadas em diversos problemas como: a) aproximador universal de funções, b) controle de processos, c) agrupamento de dados e entre outros (SILVA; SPATTI; FLAUZINO, 2010).

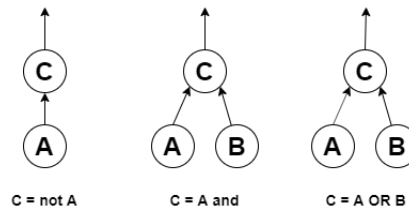
Além disso, as redes neurais artificiais são o núcleo do *Deep Learning*, sendo versáteis, potentes e escalonáveis e ideais para realizar tarefas grandes e altamente complexas de *Machine Learning*, como classificar bilhões de imagens e fornecer serviços de reconhecimento de fala (GÉRON, 2017).

2.4.1 Perceptron

O modelo de neurônio mais simples foi proposto por McCulloch e Pitts (MCCULLOCH; PITTS, 1943). O neurônio artificial simplesmente ativa sua saída quando mais de um certo número de sua entrada está ativo. McCulloch e Pitts mostraram que, mesmo com um modelo tão simplificado, é possível construir uma rede de neurônios artificiais que calcula certas portas lógicas como mostra a Figura 15. (GÉRON, 2017)

Um novo modelo de neurônio foi proposto por Frank Rosenblatt (ROSENBLATT, 1958) em 1957 o mesmo era ligeiramente diferente do proposto por McCulloch e Pitts, este

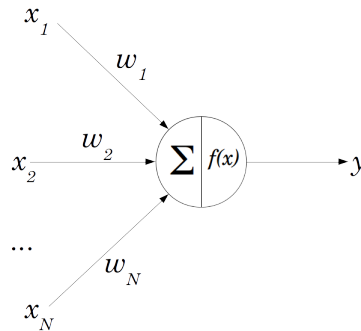
Figura 15 – Perceptron executando computação lógica simples, onde A e B são as entradas e C a saída resultante



baseia-se em um neurônio artificial, chamado unidade lógica de limiar, ou às vezes uma unidade linear de limiar, onde as entradas e saídas são números.

A descrição matemática do Perceptron, como mostra a Figura 16, é composta por valores x_1, \dots, x_n (ativações – entradas do neurônio), uma saída y (representa o axônio), os terminais de entrada têm pesos acoplados w_1, \dots, w_n (ponderações exercidas pelas junções sinápticas) e por último tem o limiar de ativação ou bias b , que é uma variável de entrada, porém não é ponderada por peso algum. O Perceptron “dispara” (quando a saída y for igual a 1) quando a soma das entradas ponderadas com seus devidos pesos e o bias ultrapassa o limiar de excitação, conforme visto na Equação 2.7, essa limiar é obtido através da aplicação de uma função de ativação (BRAGA; FERREIRA; LUDERMIR, 2007).

Figura 16 – Representação gráfica de um Perceptron



$$y = \sum_{k=1}^N x_i w_i + b \quad (2.7)$$

Pode-se escrever a mesma Equação acima em termos vetoriais como mostra a Equação 2.8, (os vetores serão escritos em negrito):

$$\mathbf{w} = (w_0, w_1, \dots, w_n)$$

$$\mathbf{x} = (x_0, x_1, \dots, x_n)$$

$$y = \mathbf{W}^T \mathbf{X} + b \quad (2.8)$$

Função de ativação A função de ativação é responsável por gerar a saída y do neurônio, estas podem ser divididas em dois grupos: funções parcialmente diferenciáveis e totalmente diferenciáveis (SILVA; SPATTI; FLAUZINO, 2010).

a) Parcialmente diferenciáveis: Cujas derivadas de primeira ordem são inexistentes em alguns pontos, por exemplo: função degrau, Equação (2.9) ilustrada pela Figura 17 e a função rampa simétrica, Equação (2.10) ilustrada pela Figura 18.

Degrau:

$$u(t) = \begin{cases} 1, & \text{se } t \geq 0 \\ 0, & \text{se } t < 0 \end{cases} \quad (2.9)$$

Rampa (onde “ a ” é uma constante):

$$g(u) = \begin{cases} a, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < -a \end{cases} \quad (2.10)$$

Figura 17 – Função degrau

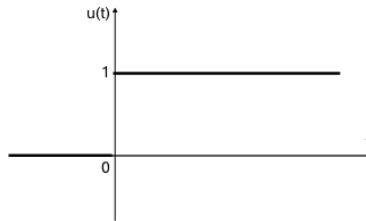
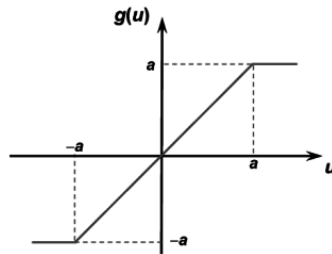


Figura 18 – Função rampa simétrica

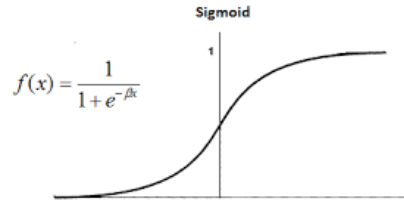


b) Totalmente diferenciáveis: Cujas derivadas de primeira ordem existem e são conhecidas em todos os pontos de seu domínio. Por exemplo: função sigmoide, Equação 2.11 ilustrada pela Figura 19 e função linear, Equação (2.12) ilustrada pela Figura 20.

Sigmoide:

$$f(x) = \frac{1}{1 + e^{-\beta x}} \quad (2.11)$$

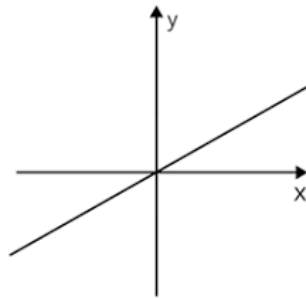
Figura 19 – Função sigmoide



Linear:

$$g(x) = x \quad (2.12)$$

Figura 20 – Função linear



Algoritmo de aprendizagem O algoritmo de aprendizagem ou algoritmo de treinamento do Perceptron pode ser resumido como: *"O algoritmo de treinamento Perceptron proposto por Rosenblatt foi amplamente inspirado pela regra de Hebb. Em seu livro - The Organization of Behavior - publicado em 1949, Donald Hebb sugeriu que quando um neurônio biológico geralmente aciona outro neurônio, a conexão entre esses dois neurônios fica mais forte. O Perceptron é treinado usando uma variante desta regra que leva em consideração o erro cometido pela rede; ele não reforça as conexões que levam à saída errada. Mais especificamente, o Perceptron é alimentado uma instância de treinamento por vez e, para cada instância, faz suas previsões. Para cada neurônio de saída que produziu uma previsão errada, ele reforça os pesos de conexão das entradas que teriam contribuído para a previsão correta."* citegeron2017hands

O algoritmo de aprendizagem supervisionado pode ser descrito matematicamente como: Inicialmente considere duas variáveis fornecidas para o treinamento do Perceptron, sendo \mathbf{x} , Γ_d , onde \mathbf{x} é o vetor de entrada e Γ_d é a saída desejada para a correspondente entrada. O primeiro passo é inicializar o vetor \mathbf{w} com valores aleatórios dentro de um intervalo definido, que pode variar de -0.5 a 0.5 por exemplo. O segundo passo é o *feed-forward*, onde os valores das entradas serão multiplicados pelo vetor de pesos e somado com o bias e o resultado desta operação será aplicada na função de ativação como mostra a Equação 2.7. O resultado da função de ativação será a saída atual do Perceptron ao vetor de entrada \mathbf{x} que será chamado de Γ , podendo-se então definir o erro como sendo: $e = \Gamma_d - \Gamma$. O objetivo do treinamento é tornar o erro o mais baixo possível (o ideal seria

erro nulo), para isso deve-se encontrar os valores do vetor \mathbf{w} que contribuam para essa solução.

Após o término do passo *feed-forward* o pesos e o bias devem ser atualizados para minimizar o erro, então o passo seguinte é utilizar a resposta do erro e alterar o vetor \mathbf{W} conforme a Equação seguinte: $\mathbf{W}(n+1) = \mathbf{W}(n) + \eta e \mathbf{X}(n)$, onde η é a taxa de aprendizagem, sendo este uma medida da rapidez com que o vetor de pesos será atualizado. Rosenblatt definiu ainda um Teorema da Convergência (ROSENBLATT, 1958), onde a atualização dos pesos leva sempre a uma solução, caso as classes em questão sejam linearmente separáveis.

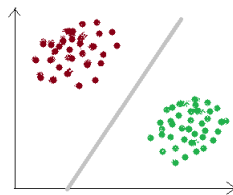
Algumas considerações sobre o treinamento:

a) O valor de η é atribuído pelo usuário, porém um valor muito baixo de η pode levar a um tempo de convergência alto e um valor de η alto pode levar a instabilidades no treinamento, foi definido que o valor ideal de η está entre 0 e 1 (ROSENBLATT, 1958)(BRAGA; FERREIRA; LUDERMIR, 2007).

b) Em relação aos valores iniciais atribuídos aos elementos do vetor de pesos, uma regra geral é iniciá-los com valores amostrados em uma distribuição uniforme definida no intervalo de $[-a, a]$, onde a é um valor próximo de zero (BRAGA; FERREIRA; LUDERMIR, 2007)(SILVA; SPATTI; FLAUZINO, 2010).

c) Separabilidade linear absoluta: Dois conjuntos A e B de pontos em um espaço n-dimensional são chamados absolutamente linearmente separáveis se $n+1$ números reais w_1, \dots, w_{n+1} existem, tal que cada ponto $(x_1, x_2, \dots, x_n) \in A$ satisfaça $\sum_{i=1}^n w_i x_i > w_{n+1}$ e cada ponto $(x_1, x_2, \dots, x_n) \in B$ satisfaça $\sum_{i=1}^n w_i x_i < w_{n+1}$. Se um Perceptron com limiar zero pode separar linearmente dois conjuntos finitos de vetores de entrada, apenas um pequeno ajuste em seus pesos é necessário para obter uma separação linear absoluta (ROJAS, 2013). Dois conjuntos finitos de pontos, A e B, no espaço n-dimensional que são linearmente separáveis também são absolutamente linearmente separáveis. A Figura 21 ilustra dois conjuntos A e B linearmente separáveis.

Figura 21 – Representação gráfica da separação linear de dois conjuntos A e B



2.5 Computação Quântica e Aprendizagem de máquina

Utilizando o computador quântico para executar algoritmos de aprendizagem de máquina, espera-se uma aceleração no tempo de execução dos algoritmos através de métodos quânticos (BIAMONTE et al., 2017). Porém, o tempo de execução é apenas uma preocupação dos algoritmos de aprendizagem, o verdadeiro desafio é formular novos algoritmos de aprendizagem de máquina que se baseiam não mais na estrutura de informação da computação clássica e sim na estrutura de informação utilizando qubits, e que possam superar não só o tempo de execução, mas as antigas limitações que os algoritmos enfrentavam e poder solucionar problemas que os computadores clássicos não conseguiam. Estas são algumas hipóteses que se pode formular baseado nas teorias da computação quântica, porém é necessário testá-las.

2.5.1 Aprendizagem de máquina quântica

Projetar um bom algoritmo quântico é uma tarefa desafiadora. Isso não deriva necessariamente da dificuldade da mecânica quântica. O problema maior está nas expectativas de que um algoritmo quântico seja mais rápido e computacionalmente menos complexo do que qualquer algoritmo clássico conhecido para a mesma finalidade (BIAMONTE et al., 2017). Estas expectativas foram atendidas em alguns algoritmos já existentes, por exemplo, o algoritmo de Grover e de Shor².

As tecnologias da computação quântica, por um lado, e o aprendizado de máquina, por outro, são ambas tecnologias emergentes que provavelmente terão um impacto transformador na nossa sociedade no futuro. As respectivas tecnologias descritas têm suas próprias questões e desafios específicos, que até agora foram investigados de maneira bastante independente. No entanto, em um corpo crescente de trabalhos recentes, os pesquisadores têm investigado a questão de até que ponto essas tecnologias podem realmente aprender e se beneficiar um do outro, sendo possível separar o modo como estas podem se relacionar uma com a outra em 4 tipos de vertentes, sendo o que diferencia cada uma destas vertentes são: o modo como um algoritmo é formulado (se foi baseado em um matemática quântica ou não) e o tipo de dado (dado provém de informação clássica ou quântica), como pode ser visto na Figura 22 (DUNJKO; BRIEGEL, 2018). Existem 4 maneiras distintas de combinar o tipo de dado e o tipo de algoritmo utilizado em clássico e quântico, de cima para baixo e da esquerda para direita, estas maneiras podem ser descritas como:

A) É possível existir um algoritmo clássico que utiliza dados clássicos que é o caso da aprendizagem de máquina clássica;(DUNJKO; BRIEGEL, 2018).

B) Existem algoritmos quânticos que lidam com dados clássicos que é o caso da aprendizagem de máquina quântica;(DUNJKO; BRIEGEL, 2018).

² O apêndice aborda como esses algoritmos funcionam.

C) Algoritmos clássicos podem ser usados com dados quânticos; (DUNJKO; BRIEGEL, 2018).

D) Por último existem algoritmos que lidam com dados quânticos (DUNJKO; BRIEGEL, 2018).

Aprendizagem de máquina quântica é uma vertente que explora a interação entre computação quântica e aprendizagem de máquina, investigando como os resultados e técnicas de um campo podem ser usadas para resolver os problemas do outro. Recentemente, tem-se testemunhos de avanços significativos em ambas as direções de influência. Por exemplo, na computação quântica está se encontrando uma aplicação vital para o aceleração do tempo de treinamento para os algoritmos de aprendizagem de máquina, tal aplicação é crítica no mundo que vivência o "big data". Por outro lado, a aprendizagem de máquina já permeia muitas tecnologias de ponta e pode tornar-se instrumental em tecnologias quânticas avançadas (DUNJKO; BRIEGEL, 2018).

Figura 22 – Diferentes tipos de vertentes que unem aprendizagem de máquina e computação quântica. A diferença entre elas é a forma como o algoritmo é descrito e como o dado é tratado.

		Tipo de Algoritmos	
		clássico	quântico
Tipo de Dados	clássico	CC	CQ
	quântico	QC	QQ

Pode-se mostrar duas razões pela qual um especialista em aprendizagem de máquina (AM) deva estar interessado em computação quântica. Primeiro, com uma crescente quantidade de dados, os sistemas atuais de AM estão aproximando-se rapidamente dos limites dos modelos computacionais clássicos. Nesse sentido, os algoritmos quânticos oferecem soluções mais rápidas para processar informações para classes de problemas selecionadas. Segundo, os resultados da teoria da aprendizagem quântica apontam, sob certas suposições, uma separação comprovável entre a aprendizagem clássica e a aprendizagem quântica. Isso implica que problemas clássicos difíceis podem ser resolvidos com a adoção de paradigmas computacionais baseados na computação quântica. Entretanto, o otimismo deve vir com uma dose de ceticismo (CILIBERTO et al., 2018).

Os algoritmos quânticos conhecidos para problemas de aprendizagem de máquina sofrem com uma série de advertências que limitam suas práticas e aplicabilidade e, até o momento, ainda não é possível concluir que os métodos quânticos terão um impacto significativo na aprendizagem de máquina, isso se deve ao fato que os algoritmos quânticos estão mais a frente do que o desenvolvimento e aperfeiçoamento dos hardwares quânticos, porém espera-se que no futuro tais algoritmos não sofram mais estas limitações (CILIBERTO et

al., 2018).

Neste trabalho será abordado um experimento que visa demonstrar um novo algoritmo de aprendizagem de máquina quântico que possa contribuir com a ideia de que no futuro a computação quântica possa ter um impacto significativo na melhora dos algoritmos de aprendizagem de máquina. Esse novo algoritmo implementa um Perceptron fundamentado nos princípios da computação quântica, sendo o tipo de vertente se encaixando em um algoritmo quântico que utiliza dados clássicos, como pode ser visto na Figura 22. A ideia é que no futuro seja possível evoluir para uma rede neural quântica que consiga superar a performance das redes neurais clássicas.

2.5.2 Limitação do Perceptron clássico

Para entender o contexto do porquê do desenvolvimento de um novo tipo de Perceptron quântico, é preciso apresentar a limitação que o Perceptron de Rosenblatt enfrenta.

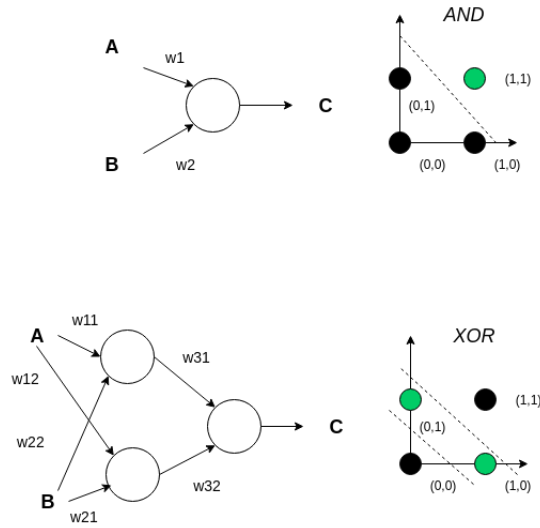
O Perceptron de Rosenblatt consegue resolver problemas linearmente se-paráveis, como obter a solução da tabela verdade³ AND e OR. Já quando não trata-se somente de um problema linearmente separável, como, a tabela verdade da XOR, é preciso aumentar o número de Perceptrons para encontrar a solução, a maneira mais simples e eficaz é adicionar uma camada oculta (YANLING; BIMIN; ZHANRONG, 2002), onde nesta serão inseridos dois Perceptrons. Na última camada um terceiro Perceptron é responsável por unir o resultado da camada anterior e mapear este resultado em uma saída binária única, como mostra a Figura 23. A medida que o problema torna-se mais complexo o número de Perceptrons vai aumentando gradativamente. Contudo, o aumento do número de Perceptrons acarreta em uma quantidade maior de processamento matemático a ser feito pela máquina, então o problema agora reside no poder computacional da máquina, onde o fator limitante para solucionar os problemas é de sua responsabilidade. Melhorar o poder computacional da máquina acarreta em um maior o custo financeiro (deve-se comprar mais hardwares) e energético (casto maior de consumo energético para manter a máquina em operação) para suprir a demanda.

Essa limitação do Perceptron têm motivado muitos pesquisadores a investigar novas tendências no domínio da computação neural (SHAFEE, 2007). Uma dessas novas tendências é a aplicação de propriedades e técnicas da computação quântica nas abordagens clássicas da computação neural.

Portanto, a hipótese inicial deste trabalho é conseguir solucionar o problema da limitação do Perceptron clássico empregando uma abordagem baseada nas teorias da computação quântica, uma vez que o mesmo não é capaz de resolver os problemas

³ é uma ferramenta de natureza matemática que tem como objetivo verificar a validade lógica de uma proposição composta (argumento formado por duas ou mais proposições simples).

Figura 23 – Problema linearmente separável e não separável.



linearmente não-separáveis e acaba por propiciar um aumento gradativo no uso de neurônios quando o problema é considerado mais complexo.

Alguns trabalhos foram realizados embasados nesta mesma hipótese, estes acabaram propondo novas técnicas de adaptação do Perceptron para a computação quântica, sendo alguns bem pertinentes, como no caso das pesquisas de Sagheer e Zidan ([SAGHEER; ZIDAN, 2013](#)) que apresentam uma nova abordagem computacional de um Perceptron quântico que obtém um aprendizado de baixo custo computacional, onde o neurônio seria capaz de construir operadores de ativação auto-adaptáveis, capazes de realizar o processo de aprendizado em um número limitado de iterações e, assim, reduzir o custo computacional geral. Os operadores de ativação propostos por Sagheer e Zidan são concebidos como um conjunto de M operadores onde cada operador de ativação é uma matriz de rotação que depende de dois parâmetros: um ângulo ϕ e um ângulo θ . Porém, para cada entrada do Perceptron quântico de Sagheer e Zidan, um operador de ativação é utilizado, ou seja, caso tenha N entradas, N operadores de ativação são utilizados. Portanto, o Perceptron quântico idealizado por Sagheer e Zidan não é muito funcional para um conjunto de entradas muito grandes, já que muitos operadores de ativação serão necessários, além disso Sagheer e Zidan não propuseram um circuito quântico que seja capaz de realizar o algoritmo proposto por eles, ou seja, na teoria eles conseguiram mostrar um avanço frente ao Perceptron clássico, porém a implementação em um computador quântico não foi discutida por estes.

Todavia, a ideia do conceito de computação neural quântica teve início em 1995, com Kak ([KAK, 1995](#)), que foi o primeiro pesquisador a introduzir o conceito de computação neural quântica. Porém, foi em 2001, com o cientista chamado Altaisky ([ALTAISKY, 2001](#)) que foi desenvolvido um Perceptron quântico simples, sendo este bem similar ao Perceptron clássico, porém Altaisky propôs a utilização de uma função de ativação quântica. Esta

função não foi bem definida por ele e embora tenha sido usado o formalismo da matemática quântica para representar o modelo clássico de Perceptron, é difícil estender essa proposta a um modelo de rede neural completo.

Em seguida, outro pesquisador chamado Fei ([FEI; BAOYU, 2003](#)) introduziu um novo modelo de neurônio quântico, onde seu algoritmo de aprendizado foi amparado no Perceptron de Altaisky. O modelo de Fei utilizou a regra delta como a regra de aprendizado, que acabou por produzir resultados consideráveis, como calcular a função XOR usando apenas um neurônio. Infelizmente, o modelo de Fei não forneceu uma nova maneira de derivar o operador de ativação, bem como no modelo de Altaisky e Sagheer e Zidan, a maneira de substituir a função de ativação clássica para uma quântica, acabou por não facilitar a construção e implementação de um circuito quântico, no qual fosse possível mostrar sua aplicabilidade em um computador quântico.

3 Desenvolvimento, resultados e discussões

3.1 Desenvolvimento

Diante da limitação do Perceptron clássico e da dificuldade de re-formular o mesmo algoritmo para uma nova versão baseada nos princípios da computação quântica e construir um circuito que seja capaz de demonstrar a aplicabilidade deste em um computador quântico, foram considerados os seguintes argumentos:

a) Diferentemente do que Altaisky propôs ao depender o processamento do seu Perceptron quântico em uma seleção de um operador de ativação, a antiga função de ativação do Perceptron clássico poderia ser reformulada, onde agora em sua nova versão quântica a operação que realizava o mapeamento das entradas em valores booleanos de saída, seria modificada para quaisquer valores de saída, pois um qubit pode assumir diferentes valores ao mesmo tempo graças a sua propriedade de superposição, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Desta forma apenas algumas porta quânticas seriam necessárias para formular uma nova função de ativação.

b) Os pesos do Perceptron clássico iram se manter, porém ao invés de descartá-los como Sagheer e Zidan realizaram, seria possível adaptar os pesos em uma grandeza que pudesse alterar a probabilidade de medição de um qubit e assim simular o que o Perceptron de Rosenblatt realizava. Essa grandeza seria a fase de um qubit onde portas de rotação podem ser usadas para modificar as fases destes.

c) A melhoria na velocidade de processamento no computador quântico permitiria um número menor de interações para chegar no valor desejado de pesos.

Partindo disso, a complexidade agora seria formular um circuito quântico que unisse estas ideias. Todavia, é necessário apresentar um novo algoritmo de treinamento do Perceptron quântico, porém este poderia ser uma adaptação do seu antecessor, modificando agora a forma como os dados vão ser tratados e como o novo peso vai ser atualizado.

3.1.1 Algoritmo de aprendizagem do Perceptron Quântico

O algoritmo de aprendizagem do Perceptron quântico segue os fundamentos da regra de atualização de Hebb, onde este vai reforçar os pesos no momento que o neurônio produzir uma saída errada. Sendo que os pesos no Perceptron clássico são representados por números reais, porém agora no Perceptron quântico ou QPerceptron os pesos são representados pela grandeza da fase do qubit, ou seja, são representados por valores inteiros (valores de ângulos) que variam de 0° a 180° . Para atualizar os novos pesos quânticos é

preciso mudar a fase, ou seja, "rotacionar o qubit" e após essa rotação é preciso medir o valor da mesma. Visto que no postulado 3 da mecânica quântica apenas os valores da amplitude podem ser medidos, é preciso então realizar uma mudança para que o valor da fase apareça no cálculo de medição. Felizmente, isso é possível graças a uma mudança na base computacional utilizando uma porta quântica, isto será visto na próxima seção, além disso vale ressaltar uma vantagem de utilizar a fase como peso, que é a existência de uma porta quântica que consegue modificar o ângulo de fase do qubit, esta porta quântica é chamada de rotação de fase em Z ou R_ϕ .

Peso Quântico A porta quântica que permite a formulação do peso quântico pode ser definida como:

$$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

A porta deixa o vetor de base $|0\rangle$ inalterado, enquanto mapeia o vetor base $|1\rangle$ para $e^{i\theta}|1\rangle$, a porta pode ser ilustrado conforme a Figura 24.

Figura 24 – Operação da Porta quântica R_ϕ .

$$\beta|1\rangle + \alpha|0\rangle \longrightarrow \boxed{R_{\phi_n}} \longrightarrow \alpha|0\rangle + \beta e^{i\phi}|1\rangle$$

A probabilidade de medição é alterada se for feito uma mudança da base computacional para outro tipo de base conforme visto na fundamentação teórica.¹ Relembrando a conversão para a nova base de medição e mostrando que é possível utilizar o valor da fase como medição, a Equação abaixo descreve essa situação:

$$|\psi\rangle = \frac{1 + e^{i\theta}}{\sqrt{2}}|+\rangle + \frac{1 - e^{i\theta}}{\sqrt{2}}|-\rangle$$

O seguinte circuito, mostrado na Figura 25, permite com que a probabilidade de medição de um qubit seja descrita pela sua fase e não pela sua amplitude. O circuito é composto de duas portas Hadamard que são fundamentais para alterar a base computacional e permitir a medição por fase, e a porta R_ϕ que irá alterar a probabilidade de medição do qubit.

Figura 25 – Circuito quântico para rotacionar e trocar a medição da probabilidade de um qubit de amplitude para fase.

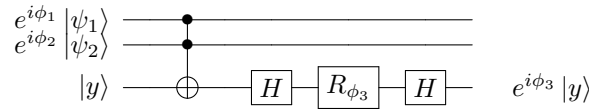
$$\beta|1\rangle + \alpha|0\rangle \longrightarrow \boxed{H} \longrightarrow \boxed{R_\phi} \longrightarrow \boxed{H} \longrightarrow \frac{1+e^{i\theta}}{\sqrt{2}}|0\rangle + \frac{1-e^{i\theta}}{\sqrt{2}}|1\rangle$$

¹ No postulado 3 foi discutido a propriedade de como alterar a mudança de medição para fase.

A primeira porta Hadarmad é responsável por alterar a base computacional $|0\rangle$ e $|1\rangle$ para $|+\rangle$ e $|-\rangle$, a porta R_ϕ rotaciona a fase em ϕ graus, e a última porta Hadarmad é responsável por alterar a base $|+\rangle$ e $|-\rangle$ de volta a base computacional.

Função de ativação A função de ativação não é a mesma do Perceptron clássico, esta era definida por funções como sigmoide, linear, degrau etc. Agora a função de ativação é uma CNOT (ou CnNOT conforme aumenta o número de entradas) junto com o circuito de mudança de probabilidade de medição por fase, ou seja, duas portas Hadamard e uma R_ϕ . A função de ativação terá uma variável de ângulo que pode ser 0° ou 180° que é aplicada na porta quântica R_ϕ , esta variável é atualizada conforme o algoritmo de treinamento. A função de ativação é responsável por transformar as variáveis de entrada em um valor de qubit $|0\rangle$ ou $|1\rangle$ e nega-los caso o valor de ângulo seja 180° . O circuito de função de ativação mostrado na Figura 26 é descrito para o caso de duas entradas.

Figura 26 – Circuito quântico para função de ativação de dois qubits.



Agora que foi descrito como irá funcionar o peso quântico e a função de ativação é possível mostrar o algoritmo de aprendizagem, já que a forma como este foi elaborado segue o modelo descrito por Ronsenblatt. O algoritmo de aprendizagem pode ser descrito matematicamente da seguinte forma: Inicialmente considera-se duas variáveis quânticas fornecidas: $|X\rangle$ e Y_d , onde $|X\rangle$ é o vetor de entrada com n entradas quânticas e Y_d é o vetor da saída desejada, o algoritmo então é descritos nos passos abaixo:

$$|X\rangle = [|x_1\rangle, |x_2\rangle, |x_3\rangle, \dots, |x_n\rangle], \text{ e}$$

$$\mathbf{Y} = [y_1, y_2, y_3, \dots, y_n].$$

1) Inicializar os pesos de fases $\phi_1, \phi_2, \dots, \phi_n$ de forma aleatória com dois possíveis valores 0° ou 180° ², e a variação para cada peso é definida como $\Delta \mathbf{W}_n$ ³ que vai ser inicializado com 0 e o bias inicializado com $|0\rangle$.

2) O passo "feed-forward" é executado seguindo as operações contidas no circuito quântico, onde a saída será $|y\rangle$ na qual este vai ser medido e assim de acordo com o postulado 3, o estado quântico de $|y\rangle$ vai ser colapsado no valor 0 ou 1.

² A escolha desses dois ângulos é para demonstrar que a probabilidade de medição seja um número inteiro, mas nada impede de começar com outros valores.

³ Essa variável irá receber um valor toda vez que o erro não for 0, e no final de uma época o peso irá receber o valor de seu respectivo $\Delta \mathbf{W}_n$

3) O resultado do valor medido de $|y\rangle$ vai ser comparado com o valor de saída desejado e assim calculado o erro: $e = y - y_n$

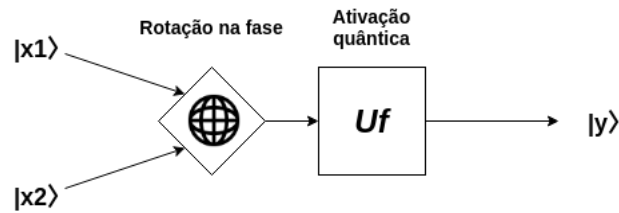
4) Após o termino do passo "feed-forward", a variação de cada peso quânticos vai ser atualizados de acordo com a Equação: $\Delta \mathbf{W}_{(n+1)} = \Delta \mathbf{W}_{(n)} + |e| |x\rangle \pi$. Só no passo 5 que os pesos irão receber o valor da váriavel $\Delta \mathbf{W}_{(n)}$. Além disso foi escolhido o valor π como a taxa de aprendizagem⁴.

5) Após a execução de todos os valores do vetor $|X\rangle$ os pesos e o bias serão atualizados conforme a regra:

- a) $\Delta \mathbf{W} \geq 360^\circ$, O ângulo da função de ativação será acrescido de 180° .
- b) $\Delta \mathbf{W} \leq 360^\circ$ e $\sum \Delta \mathbf{W} \geq 0^\circ$, O ângulo da função de ativação será acrescido de 0° .
- c) $\Delta \mathbf{W} = 0^\circ$, O Bias é negado, ou seja se for $|0\rangle \rightarrow |1\rangle$ e $|1\rangle \rightarrow |0\rangle$.

O Perceptron Quântico pode ser ilustrado conforme a Figura 27.

Figura 27 – Perceptron Quântico



⁴ Essa taxa de aprendizagem possui o mesmo objetivo do que seu antecessor clássico, o valor de π permite uma convergência mais rápida para a solução.

Algoritmo 1: Algoritmo do Perceptron quântico**Data:** $|X\rangle$, \mathbf{Y}_d , Bias = $|0\rangle$, epochs, $\Delta \mathbf{W} = 0$ **Result:** \mathbf{Y} $\phi_n \leftarrow \text{circuit}_{\text{random}}(0^\circ, 180^\circ);$ $\eta \leftarrow \pi;$ **while** $|X\rangle$ **do** $|y\rangle \leftarrow \text{circuit}_{\text{feedforward}}(|x_n\rangle, \text{Bias});$ $y_n \leftarrow \text{Measurement}(|y\rangle);$ $e \leftarrow y_n - y_d;$ $\Delta \mathbf{W}(n+1) \leftarrow \Delta \mathbf{W}(n) + |e| |x_n\rangle \eta;$

Switch()

if $\Delta \mathbf{W} \geq 360^\circ$ **then** $\phi_{\text{active-function}} \leftarrow \phi_{\text{active-function}} + 180^\circ;$ **if** $\Delta \mathbf{W} \leq 360^\circ$ **and** $\Delta \mathbf{W} \geq 0^\circ$ **then** $\phi_{\text{active-function}} \leftarrow \phi_{\text{active-function}} + 0^\circ;$ **if** $\Delta \mathbf{W} = 0^\circ$ **then** $|\text{Bias}\rangle = \text{XNOT}(|\text{Bias}\rangle)$ **if** $e \neq 0$ **then**

Retorna ao While;

else

END;

Gerando ângulos aleatórios Dada uma moeda com uma distribuição de probabilidade desconhecida, existem duas abordagens para decidir se a moeda é justa. A primeira abordagem é examinar a própria moeda; espera-se que uma moeda de formato uniforme produza resultados justos. A segunda abordagem é realmente lançar a moeda várias vezes para observar se a saída está correta. Nesta abordagem, a moeda é vista como uma caixa preta (TAMURA; SHIKANO, 2019). Um gerador de números aleatórios é semelhante a uma moeda, na medida em que se espera que produza 0s e 1s imparciais e independentes. Ao contrário de uma moeda, o mecanismo físico de um gerador de números aleatórios, geralmente, é inacessível para seus usuários. Portanto, os usuários dependem de testes estatísticos para decidir a justiça do dispositivo a partir de sua saída (TAMURA; SHIKANO, 2019).

Entretanto, para um computador quântico, gerar números aleatórios é bem simples, pois mecânica quântica fornece uma fonte pronta para o comportamento aleatório, já que todo estado quântico é probabilístico e não estritamente determinístico. Assim, por definição, o computador quântico tem capacidade para gerar números aleatórios verdadeiros. O circuito que gera um número aleatório é formada simplesmente por uma porta Hadarmad mostrado na Figura 28. A matemática para provar a geração de um binário aleatório é:

Figura 28 – Porta quântica Hadarmad aplica a um qubit $|0\rangle$

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

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

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

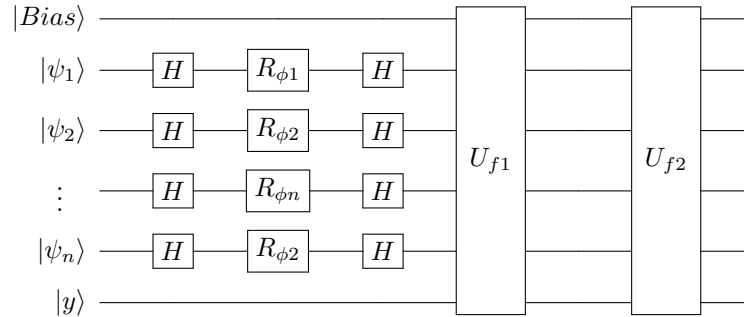
$$p(0) = \langle \psi | M_0^\dagger M_0 | \psi \rangle = \langle \psi | M_0 | \psi \rangle = \frac{1}{2}$$

$$p(1) = \langle \psi | M_1^\dagger M_1 | \psi \rangle = \langle \psi | M_1 | \psi \rangle = \frac{1}{2}$$

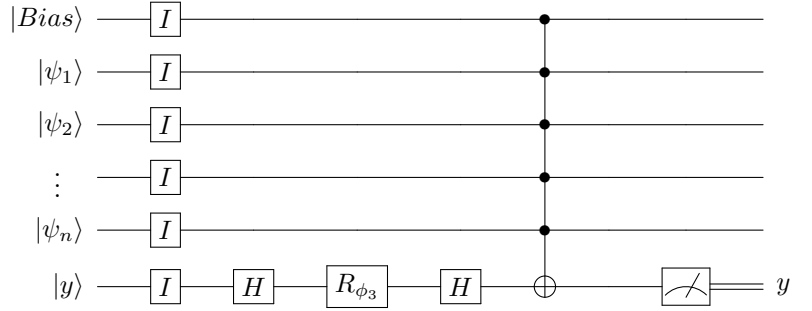
Portanto, a probabilidade de medir 0 ou 1 é 50%, ou seja, a própria natureza quântica fornece a distribuição de probabilidade correta e assim por consequência gera números verdadeiramente aleatórios.

Então, empregando o circuito de geração de número aleatório é possível obter os valores aleatórios do algoritmo do Perceptron quântico para inicializar os pesos quânticos. Desta forma, já fica evidente uma vantagem inicial do computador quântico frente ao clássico.

Passo *feed-forward* e o circuito quântico Gerados os números aleatórios para inicializar os pesos quânticos, o próximo passo é inicializar os vetores de entrada, bias e os pesos no circuito quântico *feed-forward*. O circuito quântico começa com a mudança de base e multiplicação pela matriz de rotação dos pesos para cada valor de entrada, e logo em seguida esse resultado é aplicado em dois oráculos quânticos U_{f1} e U_{f2} , este circuito quântico pode ser representado pela Figura 29 e 30. O oráculo quântico é uma forma simplificada de mostrar o funcionamento de várias portas quânticas em conjunto.

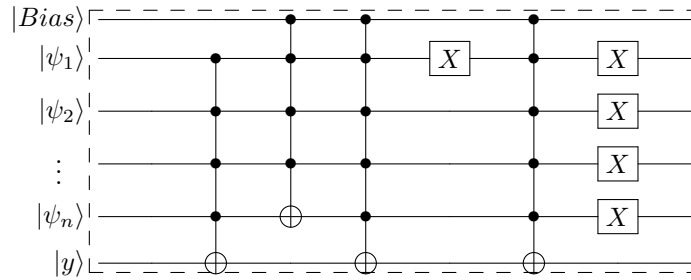
Figura 29 – Circuito quântico *feed-forward*.

A matemática que descreve o circuito não será abordado devido sua grande complexidade e extensão. Porém será mostrado abaixo como cada oráculo é por dentro. Cada

Figura 30 – Continuação do circuito quântico *feed-forward*.

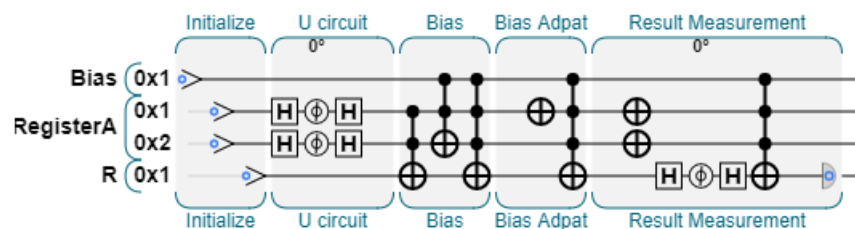
porta quântica utilizada para criar os oráculos foi discutida na fundamentação teórica e todas foram possíveis de se implementar em um computador quântico.

Oráculos: Os oráculos U_{f_1} e U_{f_2} juntos podem ser expresso com as seguintes combinações de portas quânticas mostrada na Figura 31:

Figura 31 – Representação estendida dos oráculos U_{f_1} e U_{f_2} juntos.

O circuito completo para duas entradas pode ser ilustrado na Figura 32 feita em JavaScript utilizando a plataforma web desenvolvida por (JOHNSTON; GIMENO-SEGOVIA, 2019), onde RegisterA são as duas entradas e R é o qubit de saída. O circuito é composto pelas portas quânticas: Hadamard, CNOT, X e TOFFOLI. A porta Hadamard tem como objetivo torna os qubits em superposição, a porta CNOT e TOFFOLI emaranham os qubits em superposição, enquanto que a porta X apenas troca a probabilidade dos estados $|0\rangle$ com $|1\rangle$.

Figura 32 – Circuito do Perceptron Quântico



Portanto, descrito o funcionamento do algoritmo de aprendizagem do Perceptron quântico e mostrado como é feito o circuito para implementar o passo *feed-forward*, agora será discutido e mostrado como o QPerceptron se comportou diante de duas entradas, $|\psi_1\rangle$ e $|\psi_2\rangle$, com o intuito de achar soluções para resolver todas as possíveis combinações de tabelas-verdades para a entrada binária.

3.2 Experimento

A ideia do experimento é provar a hipótese elaborada para a criação do QPerceptron, isto é, o novo algoritmo de aprendizagem do quantum Perceptron deve ser capaz de encontrar soluções para os problemas linearmente e não linearmente separáveis e também mostrar que um número menor de interações é necessário para encontrar os valores de pesos que permitem que o erro seja nulo.

O experimento foi dividido em duas partes, ambas as entradas eram binárias e as saídas desejadas correspondem aos resultados da tabela verdade das portas OR, AND, XOR, XNOR, NAND, NOR. A primeira parte do experimento consiste em testar o algoritmo em um simulador de computador quântico, este simulador está disponível como uma biblioteca chamada Qskit elaborado em Python. A segunda parte do experimento consiste em aplicar o mesmo algoritmo em 4 tipos de computadores quânticos reais, estes computadores foram desenvolvidos pela IBM e são de livre acesso.

3.2.1 Experimento 1:

O Qiskit é uma estrutura de código aberto para computação quântica ([ABRAHAM et al., 2019](#)). Ele fornece ferramentas para criar e manipular programas quânticos e executá-los em protótipos de dispositivos e simuladores quânticos. Dentro do qiskit é possível simular um computador quântico ou mandar seu programa para a IBM-Q, no qual será testado em um computador quântico real disponibilizado pela IBM.

O experimento 1 consistiu em um programa elaborado em python aplicando a biblioteca do Qiskit, que simula o algoritmo do Perceptron quântico, e dentro do programa foi possível criar todo o circuito *feed-forward* como mostra a Figura 33, nela pode ser visto que foi utilizado um qubit *strach* que é responsável por permitir a utilização da porta Toffoli de 3 entradas, já que a IBM e o Qiskit não possuem esta. Além disto os valores de o_{c0} , o_{c1} e o_{c2} correspondem a bits clássicos, este são usados para permitir que os qubits sejam medidos e assim computar na função erro.

Foram realizados os seguintes testes abaixo:

Porta OR: A porta OR consiste na seguinte tabela verdade:

Tabela 2 – tabela verdade OR

Entrada	Saída
00	0
01	1
10	1
11	1

Os dados iniciais são: o vetor de entrada $|X\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$, vetor de saídas desejadas $\mathbf{Yd} = [0, 1, 1, 1]$, e para cada possível combinação inicial de pesos, ou seja: $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$ mostrado na Figura 34, $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$ mostrado na Figura 35, $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$ mostrado na Figura 36, $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$ mostrado na Figura 37, foram obtidos os seguintes resultados:

Figura 33 – Circuito do Perceptron Quântico

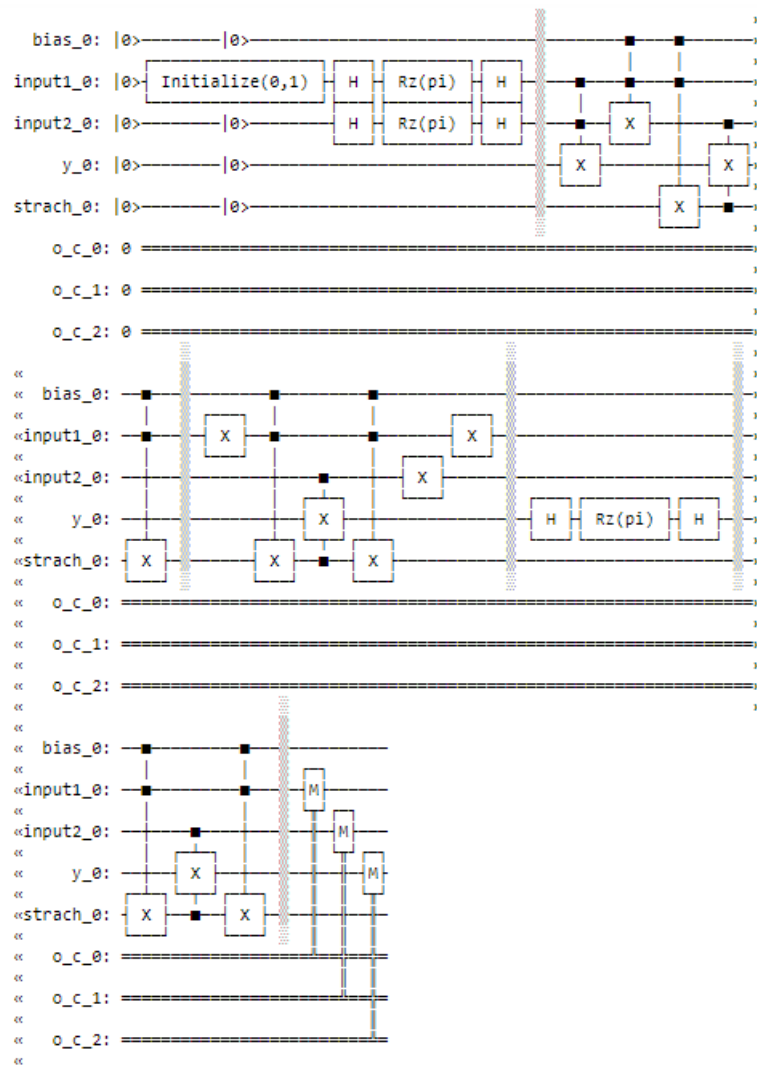


Figura 34 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 0^\circ \ \phi_2 = 0^\circ]$

```

Initialize Teta1 = 0, Teta2 = 0
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 180, Delta W2 = 180

Sum Delta W = 360 , Complete Delta W =720, erro_total = 2

Epoch = 1

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 2

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 180 , Teta2 = 180, Active = 1, Bias = 0

```

Figura 35 – Algoritmo QPerceptron para a porta OR com entrada [$\phi_1 = 180^\circ$ $\phi_2 = 0^\circ$]

```

Initialize Teta1 = 180, Teta2 = 0
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 360, Delta W2 = 180

Sum Delta W = 540 , Complete Delta W =720, erro_total = 2

Epoch = 1

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 2

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 180 , Teta2 = 180, Active = 1, Bias = 0

```

Figura 36 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 0^\circ \ \phi_2 = 180^\circ]$

```

Initialize Teta1 = 0, Teta2 = 180
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 180, Delta W2 = 360

Sum Delta W = 540 , Complete Delta W =720, erro_total = 2

Epoch = 1

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 2

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 180 , Teta2 = 180, Active = 1, Bias = 0

```

Figura 37 – Algoritmo QPerceptron para a porta OR com entrada $[\phi_1 = 180^\circ \ \phi_2 = 180^\circ]$

```

Initialize Teta1 = 180, Teta2 = 180
Epoch = 0

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 1

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 180 , Teta2 = 180, Active = 1, Bias = 0

```

A saída do simulador do computador quântico é mostrada na Figura 38, onde pode ser visto que para o vetor de entradas (eixo x) $|X\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$ a probabilidade (eixo y) do resultado de medição do qubit de saída ser 1 igual a 0.00 para apenas a entrada $|00\rangle$. O valor 0.333 indica que a probabilidade para as entradas $|01\rangle, |10\rangle, |11\rangle$ são igualmente iguais para resultar na saída 1. Se o circuito for treinado para executar a porta OR, toda vez que as entradas forem $|01\rangle, |10\rangle, |11\rangle$ a saída sempre será 1.

Também é possível ver que por exemplo, quando a entrada for $|11\rangle$ os qubits de saída do circuito para o Bias, $|\psi_1\rangle, |\psi_2\rangle$, *strach* não necessariamente devem ser permanecer iguais, pois foram aplicadas portas quânticas neste que acabaram por resulta em mudança no seu estado quântico. A Figura 39, mostra os qubits plotados na esfera de Bloch, e informa que a saída do Bias (qubit 0) permanece 0, os qubits $|\psi_1\rangle, |\psi_2\rangle$ que entraram no estado de $|11\rangle$ acabam saindo em um estado negado $|00\rangle$ (qubit 1 e 2), o *strach* (qubit 4) entrou como $|0\rangle$ e saiu como $|0\rangle$, e a saída $|y\rangle$ saiu conforme o esperado $|1\rangle$.

Figura 38 – Solução do Algoritmo QPerceptron para a porta OR no simulador da Qskit, onde para as entradas $|01\rangle$, $|10\rangle$, $|11\rangle$.

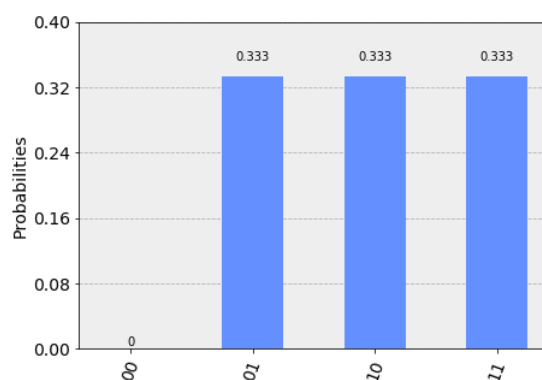
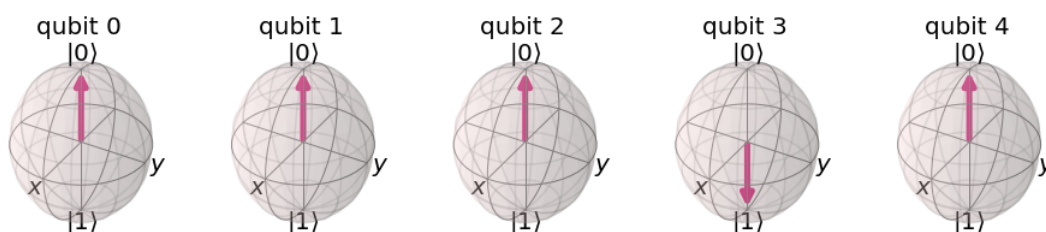


Figura 39 – Estados quânticos dos qubits de saída do Algoritmo QPerceptron para a porta OR com as entradas $|01\rangle$.



Porta AND: A porta AND consiste na seguinte tabela verdade:

Tabela 3 – tabela verdade AND

Entrada	Saída
00	0
01	0
10	0
11	1

Os dados iniciais são: o vetor de entrada $|X\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$, vetor de saídas desejadas $\mathbf{Yd} = [0, 0, 0, 1]$, e para cada possível combinação inicial de pesos, ou seja: $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$ como mostra a Figura 40, $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$ como mostra a Figura 41, $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$ como mostra a Figura 42, $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$ como mostra a Figura 43, foram obtidos os seguintes resultados:

Figura 40 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$

```

Initialize Teta1 = 0, Teta2 = 0
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 0, Inputs = 0,1
Y outuput = 0, Y desiered = 0, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 0 , Teta2 = 0, Active = 0, Bias = 0

```


Figura 41 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$

```

Initialize Teta1 = 180, Teta2 = 0
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 0, Inputs = 0,1
Y outuput = 0, Y desiered = 0, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 180, Delta W2 = 360

Sum Delta W = 540 , Complete Delta W =720, erro_total = 2

Epoch = 1

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 0, Inputs = 0,1
Y outuput = 0, Y desiered = 0, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 360 , Teta2 = 0, Active = 0, Bias = 0

```

Figura 42 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$

```

Initialize Teta1 = 0, Teta2 = 180
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 0, Inputs = 0,1
Y outuput = 1, Y desiered = 0, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 360, Delta W2 = 180

Sum Delta W = 540 , Complete Delta W =720, erro_total = 2

Epoch = 1

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 0, Inputs = 0,1
Y outuput = 0, Y desiered = 0, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 0 , Teta2 = 360, Active = 0, Bias = 0

```

Figura 43 – Algoritmo QPerceptron para a porta AND com entrada $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$

```

Initialize Teta1 = 180, Teta2 = 180
Epoch = 0

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 0, Inputs = 0,1
Y outuput = 0, Y desiered = 0, Inputs = 1,0
Y outuput = 0, Y desiered = 1, Inputs = 1,1

Delta W1 = 180, Delta W2 = 180

Sum Delta W = 360 , Complete Delta W =720, erro_total = 2

Epoch = 1

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 0, Inputs = 0,1
Y outuput = 0, Y desiered = 0, Inputs = 1,0
Y outuput = 1, Y desiered = 1, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 360 , Teta2 = 360, Active = 0, Bias = 0

```

A saída do simulador do computador quântico é mostrada na Figura 44, onde pode ser visto que para o vetor de entradas (eixo x) $|X\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$ a probabilidade (eixo y) do resultado de medição do qubit de saída ser 1 é igual a 0.00 para as entradas $|00\rangle, |01\rangle, |10\rangle$. O valor 1.00 indica que a probabilidade para a entradas $|11\rangle$ é única para que a saída resulte em 1. Se o circuito for treinado para executar a porta AND, toda vez que as entradas forem $|00\rangle, |01\rangle, |10\rangle$ a saída sempre será 0.

Também é possível ver que por exemplo, quando a entrada for $|11\rangle$ os qubits de saída do circuito para o Bias, $|\psi_1\rangle, |\psi_2\rangle$, *strach* não necessariamente devem ser permanecer iguais, pois foram aplicadas portas quânticas neste que acabaram por resulta em mudança no seu estado quântico. A Figura 45, mostra os qubits plotados na esfera de Bloch, e informa que a saída do Bias (qubit 0) permanece 0, os qubits $|\psi_1\rangle, |\psi_2\rangle$ que entraram no estado de $|11\rangle$ acabam saindo em um estado $|10\rangle$ (qubit 1 e 2), o *strach* (qubit 4) entrou como $|0\rangle$ e saiu como $|0\rangle$, e a saída $|y\rangle$ saiu conforme o esperado $|1\rangle$.

Figura 44 – Solução do Algoritmo QPerceptron para a porta AND, onde para a entrada $|11\rangle$, a probabilidade de se obter o resultado para a saída ser igual a 1 é 100% apenas para essa entrada.

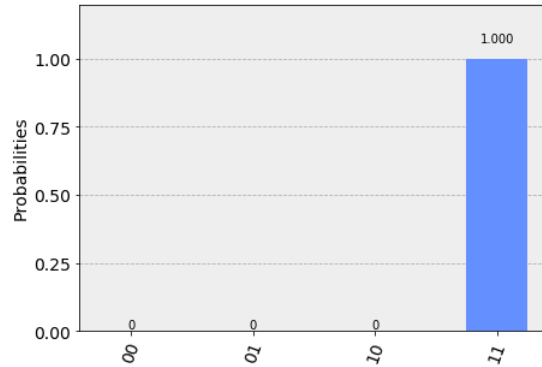
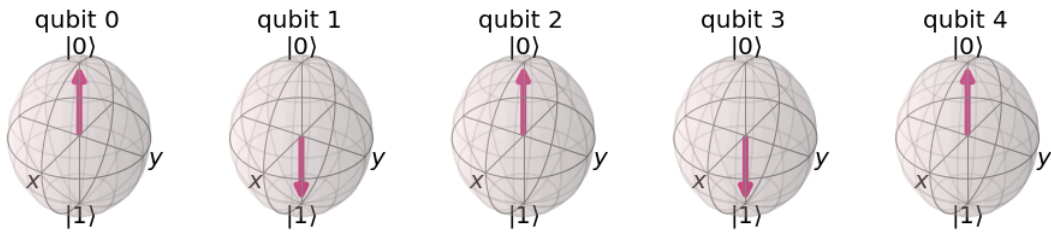


Figura 45 – Estados quânticos dos qubits de saída do Algoritmo QPerceptron para a porta OR com as entradas $|11\rangle$.



Porta XOR: A porta XOR consiste na seguinte tabela verdade:

Tabela 4 – tabela verdade XOR

Entrada	Saída
00	0
01	1
10	1
11	0

Os dados iniciais são: o vetor de entrada $|X\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$, vetor de saídas desejadas $\mathbf{Yd} = [0, 1, 1, 0]$, e para cada possível combinação inicial de pesos, ou seja: $[\phi_1 = 0^\circ \phi_2 = 0^\circ]$ como mostra a Figura 46, $[\phi_1 = 180^\circ \phi_2 = 0^\circ]$ como mostra a Figura 47, $[\phi_1 = 0^\circ \phi_2 = 180^\circ]$ como mostra a Figura 48, $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$ como mostra a Figura 49, foram obtidos os seguintes resultados:

Figura 46 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 0^\circ \ \phi_2 = 0^\circ]$

```

Initialize Teta1 = 0, Teta2 = 0
Epoch = 0

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 3

Epoch = 1

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 1, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 1

Epoch = 2

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 3

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 1, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 0 , Teta2 = 0, Active = 0, Bias = 1

```

Figura 47 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 180^\circ \ \phi_2 = 0^\circ]$

```

Initialize Teta1 = 180, Teta2 = 0
Epoch = 0

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 180, Delta W2 = 0

Sum Delta W = 180 , Complete Delta W =720, erro_total = 1

Epoch = 1

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 3

Epoch = 2

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 1, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 1

Epoch = 3

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 4

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 1, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 360 , Teta2 = 0, Active = 0, Bias = 1

```

Figura 48 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 0^\circ \ \phi_2 = 180^\circ]$

```

Initialize Teta1 = 0, Teta2 = 180
Epoch = 0

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 180

Sum Delta W = 180 , Complete Delta W =720, erro_total = 1

Epoch = 1

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 3

Epoch = 2

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 1

Epoch = 3

Y outuput = 1, Y desiered = 0, Inputs = 0,0
Y outuput = 0, Y desiered = 1, Inputs = 0,1
Y outuput = 0, Y desiered = 1, Inputs = 1,0
Y outuput = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 4

Y outuput = 0, Y desiered = 0, Inputs = 0,0
Y outuput = 1, Y desiered = 1, Inputs = 0,1
Y outuput = 1, Y desiered = 1, Inputs = 1,0
Y outuput = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 0 , Teta2 = 360, Active = 0, Bias = 1

```

Figura 49 – Algoritmo QPerceptron para a porta XOR com entrada $[\phi_1 = 180^\circ \phi_2 = 180^\circ]$

```

Initialize Teta1 = 180, Teta2 = 180
Epoch = 0

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 180, Delta W2 = 180

Sum Delta W = 360 , Complete Delta W =720, erro_total = 3

Epoch = 1

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 3

Epoch = 2

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 1, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 1

Epoch = 3

Y output = 1, Y desiered = 0, Inputs = 0,0
Y output = 0, Y desiered = 1, Inputs = 0,1
Y output = 0, Y desiered = 1, Inputs = 1,0
Y output = 1, Y desiered = 0, Inputs = 1,1

Delta W1 = 360, Delta W2 = 360

Sum Delta W = 720 , Complete Delta W =720, erro_total = 4

Epoch = 4

Y output = 0, Y desiered = 0, Inputs = 0,0
Y output = 1, Y desiered = 1, Inputs = 0,1
Y output = 1, Y desiered = 1, Inputs = 1,0
Y output = 0, Y desiered = 0, Inputs = 1,1

Delta W1 = 0, Delta W2 = 0

Sum Delta W = 0 , Complete Delta W =720, erro_total = 0

Sucess!

Answer: Teta1 = 360 , Teta2 = 360, Active = 0, Bias = 1

```

A saída do simulador do computador quântico é mostrada na Figura 50, onde pode ser visto que para o vetor de entradas (eixo x) $|X\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$ a probabilidade (eixo y) do resultado de medição do qubit de saída ser 1 é igual a 0.00 para as entradas $|00\rangle, |11\rangle$. O valor 1.00 indica que a probabilidade para as entradas $|10\rangle, |01\rangle$ são iguais para que a saída resulte em 1. Se o circuito for treinado para executar a porta XOR, toda vez que as entradas forem $|00\rangle, |11\rangle$ a saída sempre será 0.

Também é possível ver que por exemplo, quando a entrada for $|11\rangle$ os qubits de saída do circuito para o Bias, $|\psi_1\rangle, |\psi_2\rangle$, *strach* não necessariamente devem ser permanecer

iguais, pois foram aplicadas portas quânticas neste que acabaram por resulta em mudança no seu estado quântico. A Figura 51, mostra os qubits plotados na esfera de Bloch, e informa que a saída do Bias (qubit 0) permanece 1, os qubits $|\psi_1\rangle, |\psi_2\rangle$ que entraram no estado de $|11\rangle$ acabam saindo em um estado $|10\rangle$ (qubit 1 e 2), o *strach* (qubit 4) entrou como $|0\rangle$ e saiu como $|0\rangle$, e a saída $|y\rangle$ saiu conforme o esperado $|0\rangle$.

Figura 50 – Solução do Algoritmo QPerceptron para a porta XOR, onde para a entrada $|10\rangle, |01\rangle$, a probabilidade de se obter o resultado para a saída ser igual a 1 é 50% apenas para essa entrada.

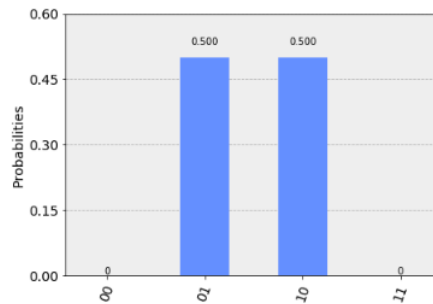
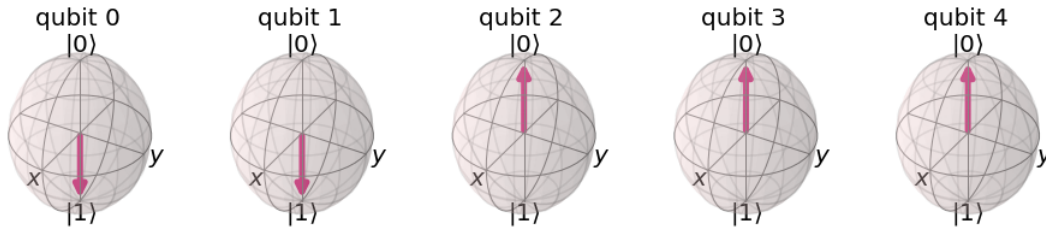


Figura 51 – Estados quânticos dos qubits de saída do Algoritmo QPerceptron para a porta XOR com as entradas $|10\rangle$.



Discussão: Como pode ser visto, o Perceptron quântico foi capaz de solucionar os problemas linearmente separáveis e não separáveis, e com poucas interações ele conseguiu encontrar os valores dos pesos quânticos ideais. Portanto, simulando o algoritmo do QPerceptron em um simulador quântico da Qiskit, os resultados foram alcançados de forma satisfatória, isto é, diferentemente das propostas pelos pesquisadores Altaisky (ALTAISKY, 2001), Fei (FEI; BAOYU, 2003), Kak (KAK, 1995) e Sagheer e Zidan (SAGHEER; ZIDAN, 2013), este algoritmo do QPerceptron foi implementado dentro de um circuito quântico e testado em um simulador, deixando claro que as propostas dos autores citados acima não deixa de ser inválido apenas por não simular e mostrar a implementação na forma de um circuito quântico. O código e um material de suporte para melhor compreensão de como foi programado e simulado o algoritmo do QPerceptron está disponível no github do autor: <https://github.com/samuelamico/QuantumPrograms>.

3.2.2 Experimento 2:

Com o sucesso do experimento 1, foi demonstrado que a teoria e a simulação do algoritmo do Perceptron quântico conseguiram alcançar os resultados esperados. Todavia, para validar o mesmo é necessário aplicar a um computador quântico real, e para isso a IBM-Q Experience ([CROSS, 2018](#)) disponibiliza 4 tipos de processadores quânticos com diferentes arquiteturas, quantidades de qubits e erros gerados com a utilização de operações quânticas diferentes.⁵

Figura 52 – Arquiteturas do IBM-Q

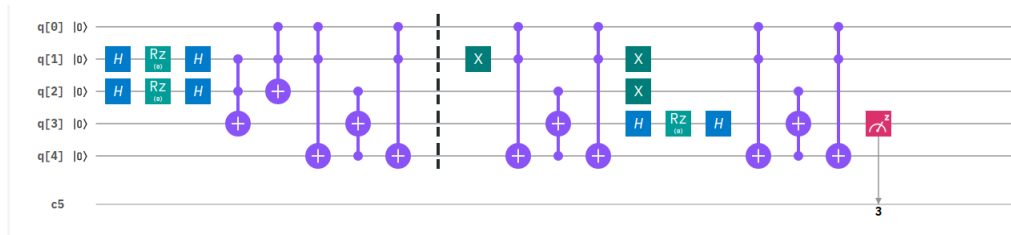


O IBM-Q Experience é uma plataforma online que fornece aos usuários em geral o acesso a um conjunto de protótipos de processadores quânticos desenvolvidos pela IBM. O IBM-Q é um exemplo de computação quântica embaçado na nuvem, onde os usuários interagem com um processador quântico através do modelo de computação quântica, aplicando portas quânticas nos qubits e usando uma GUI chamada compositor quântico, este pode ser desenvolvido utilizando o código na linguagem Assembly quântico ou através do Qiskit ([ABRAHAM et al., 2019](#)). Os processadores quânticos da IBM são compostos de qubits supercondutores de transmon, localizados em um refrigerador de diluição na sede da IBM Research em Thomas J. Watson Research Center, EUA. ([SANTOS, 2017](#)) ([CROSS, 2018](#))

⁵ O ruído quântico não é o foco deste trabalho, porém este tipo de ruído é um dos fatores que causam resultados inesperados nos experimentos quânticos reais.

No período que estava sendo escrito este trabalho, a IBM-Q Experience disponibilizou as seguintes arquiteturas de processadores quânticos descritas na Figura 52. O circuito para ser realizados pelos processadores da IBM-Q teve uma ligeira diferença em relação ao projetado na teoria, pois visto que uma porta Toffoli de 3 entradas não está disponível no catálogo, foi preciso usar um qubit de *strach* e combinar 3 portas CNOT para realizar uma Toffoli de 3 entradas, apesar dessa mudança no circuito, isto não altera em nada a forma como o circuito e o algoritmo funcionam, isso é apenas outra maneira de reescrever o circuito. A Figura 53 expõe o circuito do algoritmo de QPerceptron dentro dos processadores quânticos da IBM-Q.

Figura 53 – Circuito do QPerceptron dentro do IBM-Q

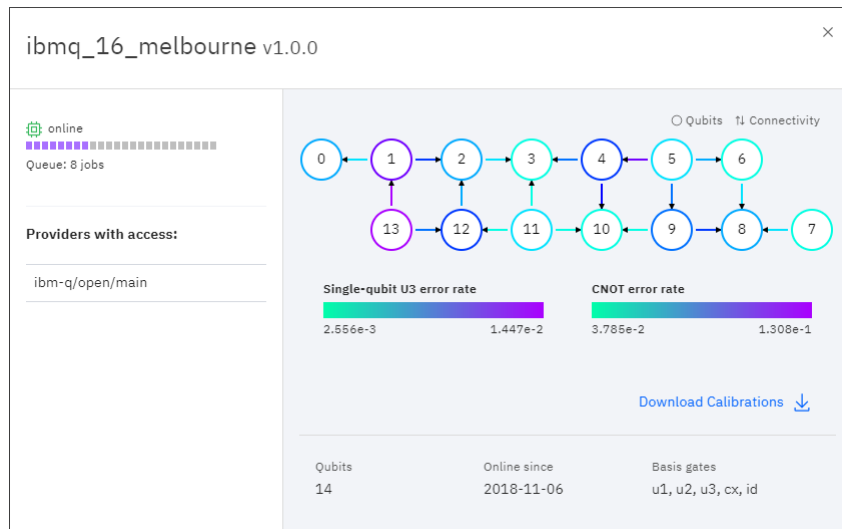


Para processar o algoritmo do QPerceptron nos processadores quânticos da IBM-Q foi necessário converter o código do experimento 1, desenvolvido em Python para a linguagem Assembly quântico. A desvantagem de manusear os processadores quânticos da IBM é a longa demora de processamento, pois como se trata de um projeto aberto para o público e como existem apenas 4 processadores, consequentemente há uma grande fila para utilizá-los, ou seja, sempre que é enviado o código para a IBM-Q é necessário aguardar um longo período na fila de espera. Por esta razão não foi possível testar todas as combinações que foram executadas no experimento 1. Os testes abaixo foram feitos para cada arquitetura de processador quântico disponível:

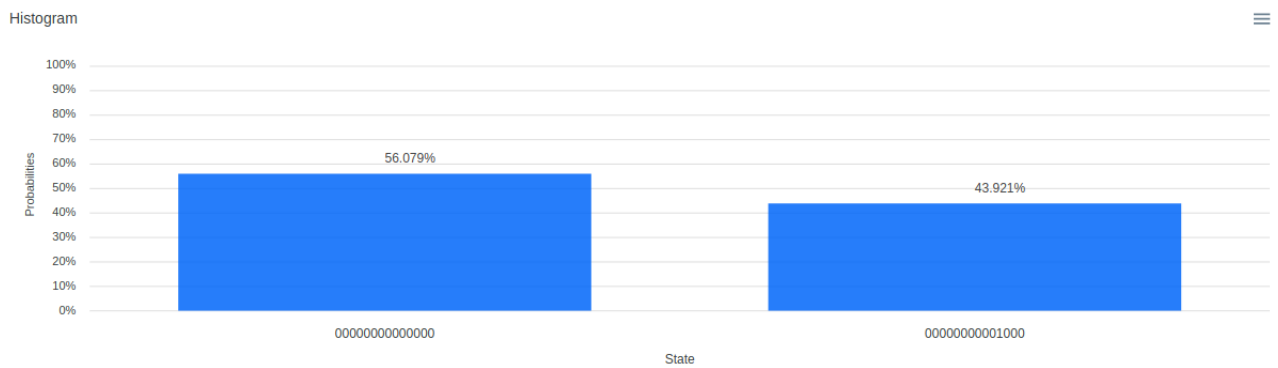
IBM-Melbourne O processador Melbourne contém 14 qubits de acesso e possui a arquitetura detalhada na Figura 54. Quando selecionado para rodar o programa dentro do IBM-Melbourne é preciso escolher o número de *trials*, quanto maior o número de *trials* maior a acurácia do experimento, entretanto a IBM-Q deixa apenas no máximo 8192 *trials*, o que teoricamente não é suficiente para obter uma boa acurácia em um experimento quântico. Para o experimento foi utilizado 5 qubits, que correspondem aos qubits 0,1,2,3,4. Conforme visto na Figura 54, o uso da porta CNOT gera um erro que varia entre $3.786e^{-2}$ e $1.308e^{-2}$.

Foi realizado o treinamento para porta AND e foi utilizado para testar o circuito do QPerceptron a entrada $|11\rangle$. O resultado da Figura 55 indica que em 8192 *trials*, 56.0796% destes apresentaram um resultado para saída igual a 0, enquanto para 43.921% do total de *trials* deu saída igual a 1.

Figura 54 – Arquitetura IBM-Melbourne

Figura 55 – Resultado para a porta AND, utilizando o processador IBM-Melbourne com entrada $|11\rangle$.

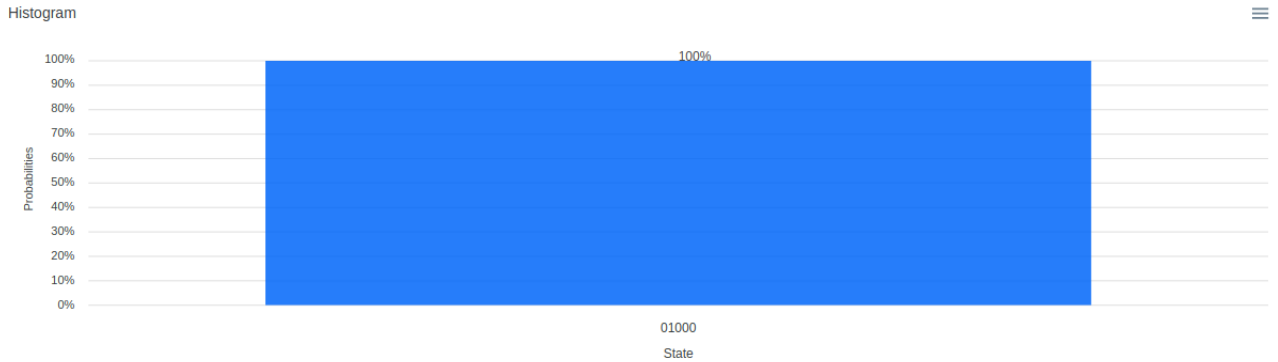
Result



A resposta para esta mesma entrada aplicada em uma simulação de um computador quântico teórico feito pela IBM corresponde a Figura 56. No simulador obteve-se que em 8192 *trials*, em 100% destes o resultado para a saída era igual a 1.

Figura 56 – Resultado para a porta AND simulado em um processador quântico teórico com entrada $|11\rangle$.

Result



IBM-Vigo O processador Vigo contém 5 qubits de acesso e possui a arquitetura como mostra a Figura 57. Testado com 8192 trials do mesmo jeito que o IBM-Melbourne. Para o experimento foi utilizado 5 qubits, que correspondem aos qubits 0,1,2,3,4. Conforme visto na Figura 57, o uso da porta CNOT gera um erro que varia entre $7.167e^{-3}$ e $1.232e^{-2}$.

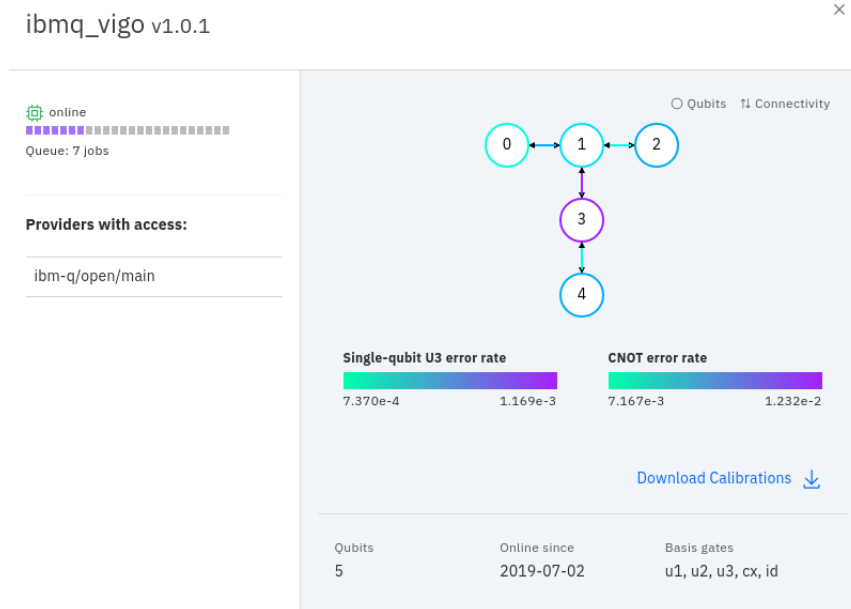
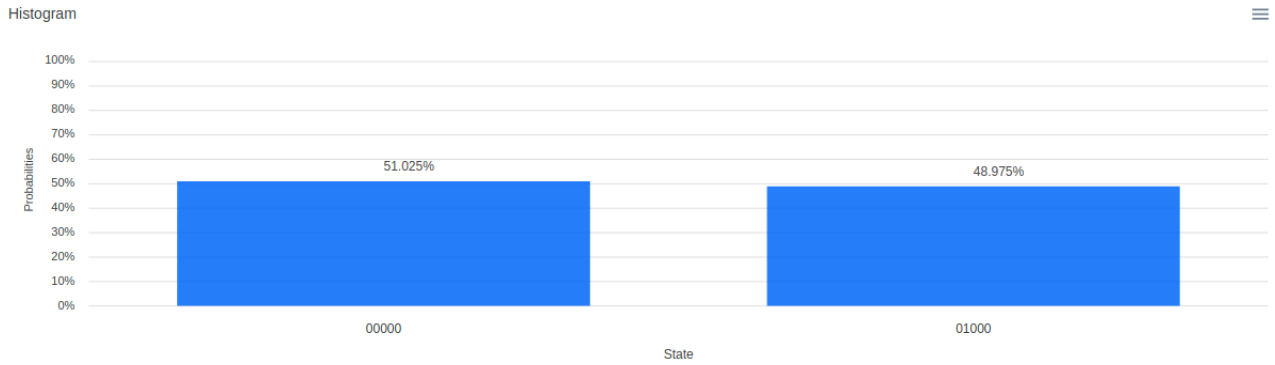


Figura 57 – Arquitetura IBM-VIGO

Foi realizado o treinamento para porta OR e foi utilizado para testar o circuito do QPerceptron a entrada $|01\rangle$. O resultado mostrado na Figura 58 indica que em 8192 trials, 51.025% destes apresentaram um resultado para saída igual a 0, enquanto para 48.975% do total de trials deu saída igual a 1.

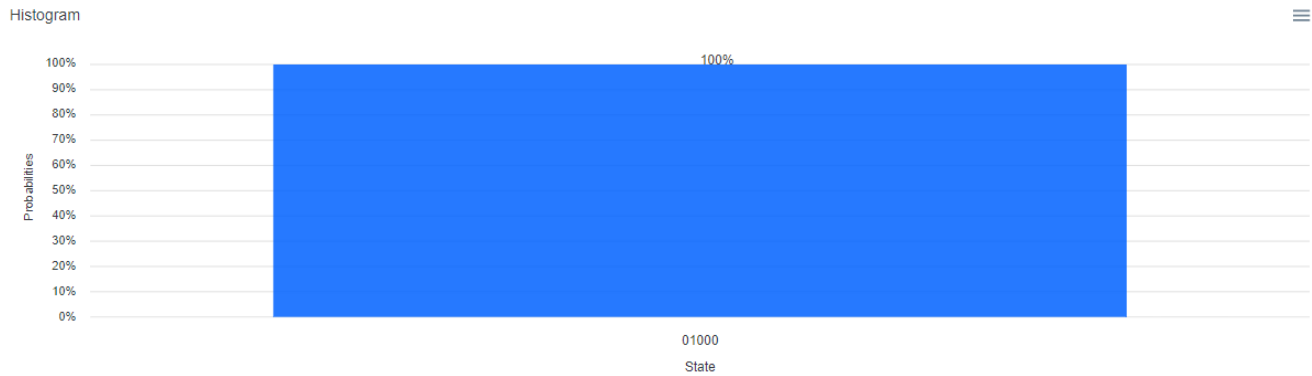
Figura 58 – Resultado para a porta OR, utilizando o processador IBM-VIGO com entrada $|01\rangle$.

Result



A resposta para esta mesma entrada aplicada em uma simulação de um computador quântico teórico feito pela IBM corresponde a Figura 59. No simulador obteve-se que em 8192 *trials*, em 100% destes o resultado para a saída era igual a 1.

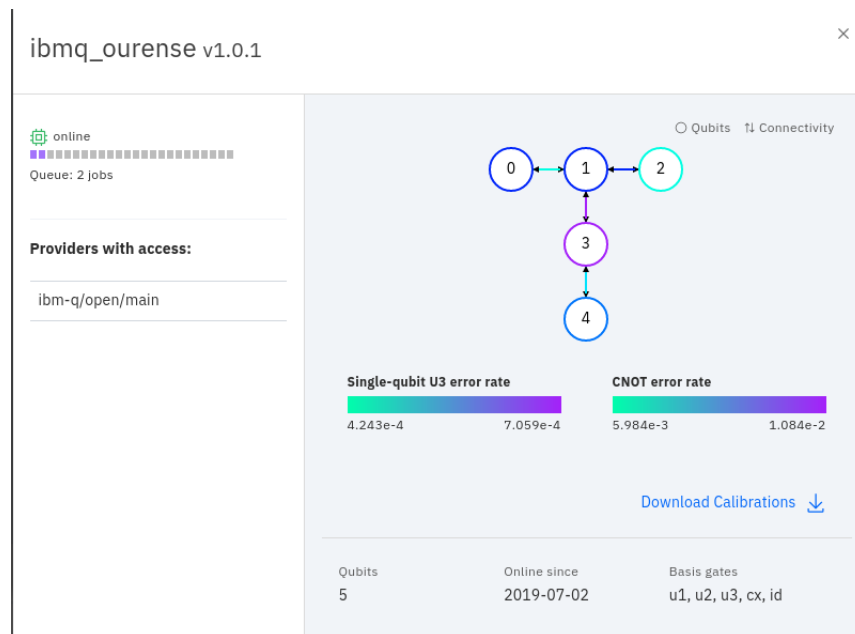
Figura 59 – Resultado para a porta OR simulado em um processador quântico teórico com entrada $|01\rangle$.



IBM-OURENSE O processador OURENSE contém 5 qubits de acesso e possui a seguinte arquitetura como mostra a Figura 60. Testado com 8192 trials do mesmo jeito que o IBM-Melbourne. Para o experimento foi utilizado 5 qubits, que correspondem aos qubits 0,1,2,3,4. Conforme visto na Figura 60, o uso da porta CNOT gera um erro que varia entre $5.984e^{-3}$ e $1.084e^{-2}$.

Foi realizado o treinamento para porta XOR e foi utilizado para testar o circuito do QPerceptron a entrada $|10\rangle$. O resultado mostrado na Figura 61 indica que em 8192

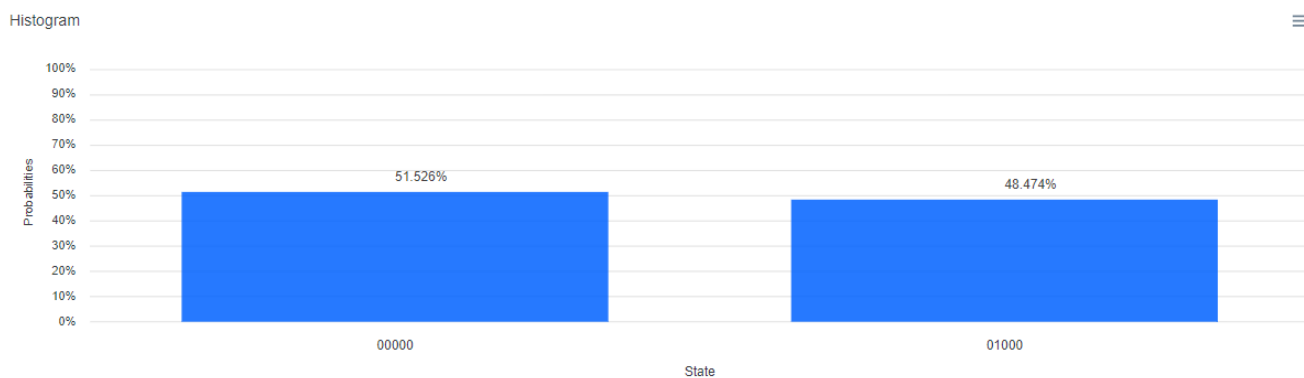
Figura 60 – Arquitetura IBM-OURENSE



trials, 51.526% destes apresentaram um resultado para saída igual a 0, enquanto para 48.475% do total de *trials* deu saída igual a 1.

Figura 61 – Resultado para a porta XOR, utilizando o processador IBM-OURENSE com entrada $|10\rangle$.

Result

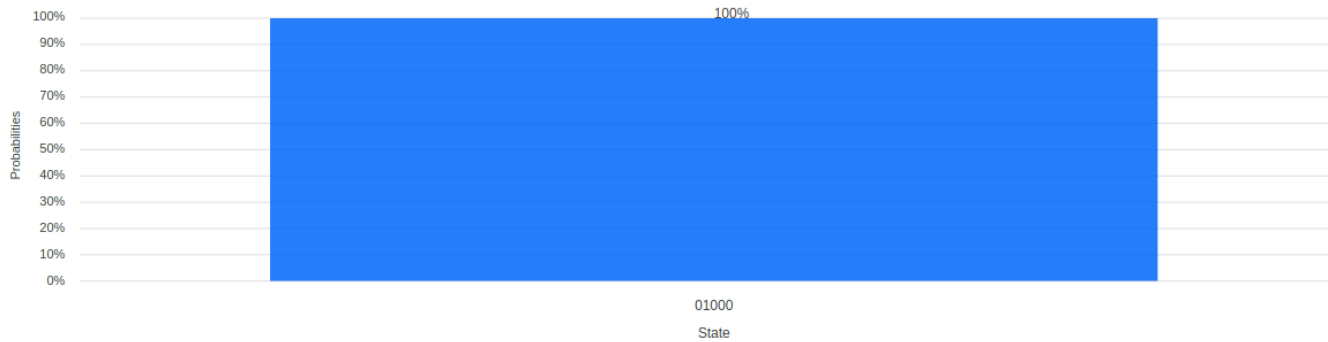


A resposta para esta mesma entrada aplicada em uma simulação de um computador quântico teórico feito pela IBM corresponde a Figura 62. No simulador obteve-se que em 8192 *trials*, em 100% destes o resultado para a saída era igual a 1.

Figura 62 – Resultado para a porta XOR simulado em um processador quântico teórico com entrada $|10\rangle$.

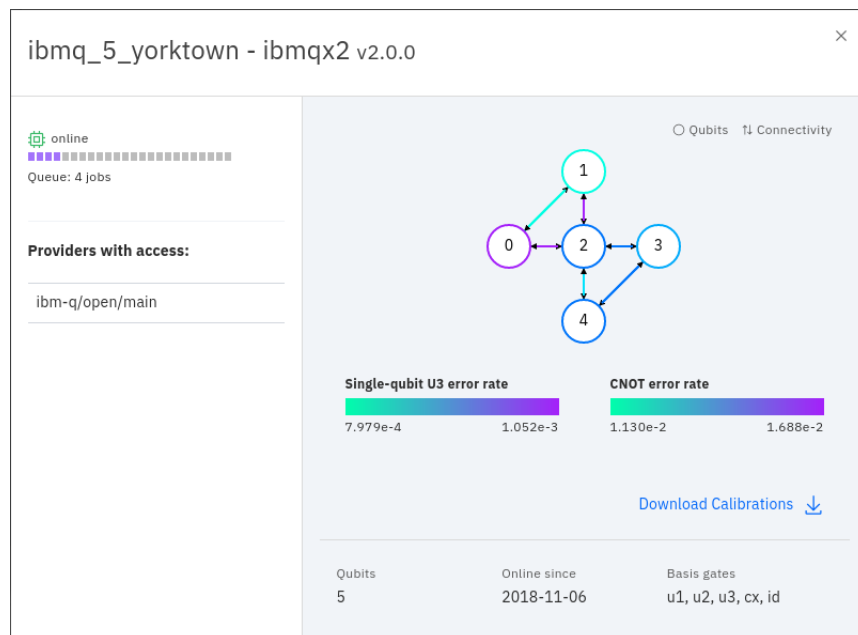
Result

Histogram



IBM-qx2 O processador qx2 contém 5 qubits de acesso e possui a seguinte arquitetura como mostra a Figura 60. Testado com 8192 trials do mesmo jeito que o IBM-Melbourne. Para o experimento foi utilizado 5 qubits, que correspondem aos qubits 0,1,2,3,4. Conforme visto na Figura 63, o uso da porta CNOT gera um erro que varia entre $1.130e^{-3}$ e $1.688e^{-2}$.

Figura 63 – Arquitetura IBM-qx2

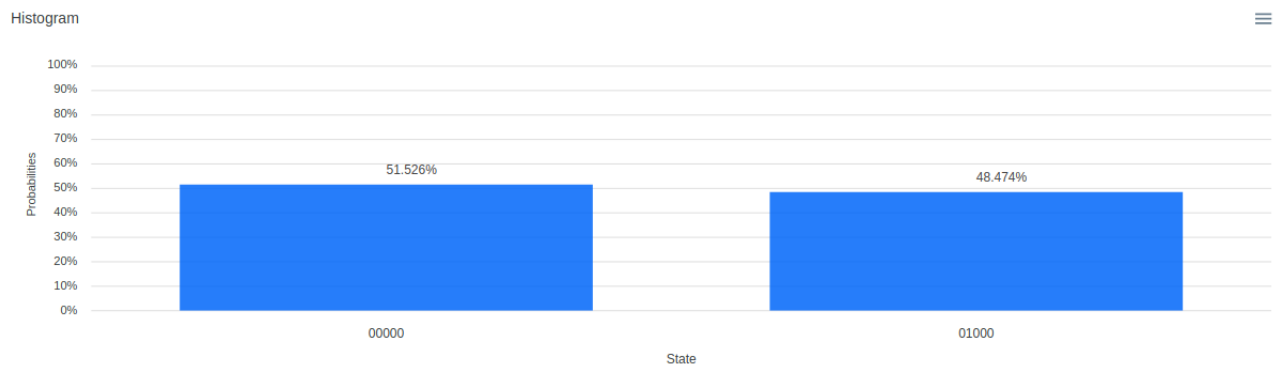


Foi realizado o treinamento para porta OR e foi utilizado para testar o circuito do QPerceptron a entrada $|00\rangle$. O resultado mostrado na Figura 64 indica que em 8192 trials,

51.526% destes apresentaram um resultado para saída igual a 0, enquanto para 48.475% do total de *trials* deu saída igual a 1.

Figura 64 – Resultado para a porta OR, utilizando o processador IBM-qx2 com entrada $|00\rangle$.

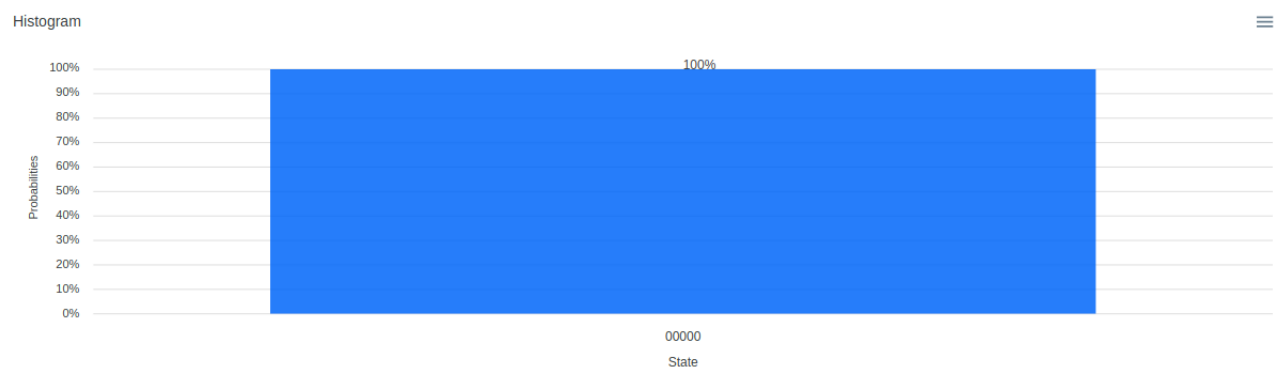
Result



A resposta para esta mesma entrada aplicada em uma simulação de um computador quântico teórico feito pela IBM corresponde a Figura 65. No simulador obteve-se que em 8192 *trials*, em 100% destes o resultado para a saída era igual a 0.

Figura 65 – Resultado para a porta OR simulado em um processador quântico teórico com entrada $|00\rangle$.

Result



Discussão

Na prática, é impossível isolar completamente um computador quântico do seu ambiente. Como todos os qubits físicos interagem com seu ambiente, os qubits computacionais de um computador quântico são adequadamente vistos como um subsistema de um sistema maior que consiste nos qubits de computação e em seu ambiente. Por ambiente, entende-se um subsistema sobre o qual não se tem controle: não se pode obter informações dele por medição ou aplicar portas a ele (RIEFFEL; POLAK, 2000).

Em alguns casos, o efeito de uma interação ambiental no subsistema computacional é reversível apenas por transformações no subsistema. Já em outros casos, ocorre decoerência. Na decoerência, as informações sobre o estado do subsistema computacional são perdidas para o ambiente. Isso é considerado grave, porque o ambiente está além do controle computacional (RIEFFEL; POLAK, 2000).

É constatável a partir dos resultados obtidos pelos diversos processadores quânticos da IBM-Q e os resultados simulados, que estes processadores tem uma margem grande de decoerência, uma vez que os seus resultados não foram tão semelhantes ao esperado. Isso acontece devido aos processadores quânticos disponíveis para projetos abertos serem protótipos que precisam de ajustes, melhor isolamento do ambiente e métodos de correção de erro, além disso o número de *trials* ainda é muito baixo.

Para que computadores quânticos práticos se aproximem muito dos resultados teóricos, são necessárias técnicas para lidar com interações do ambiente que atrapalham os cálculos quânticos e métodos de correção de erro e ruído.

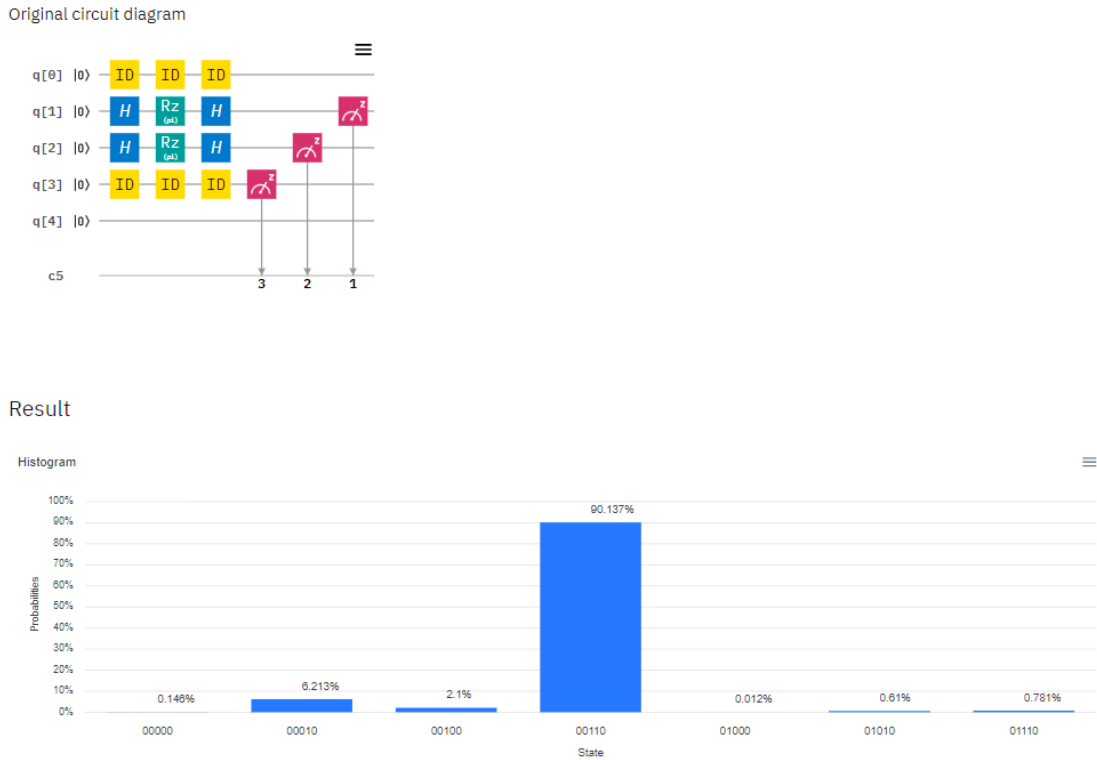
Entretanto, o trabalho não busca encontrar métodos para corrigir estes problemas nos protótipos de processadores quânticos empregados, e sim apresentar que de fato o algoritmo do Perceptron quântico é factível de ser realizado em computadores quânticos reais e que sua teoria detém vantagens de desempenho superiores as do Perceptron clássico.

3.2.3 Decoerência nos processadores quânticos da IBM Q

Abordado na discussão anterior, a existência da decoerência nos processadores quânticos da IBM Q está relacionada principalmente ao erro gerado no uso de uma porta CNOT. Na descrição de cada processador quântico da IBM Q ele informa o valor aproximado do erro ao utilizar a porta quântica CNOT entre um qubit e outro, sendo que o erro varia dependendo do qubit utilizado. Para demonstrar como este erro influencia muito nos resultados do circuito do QPerceptron, foram realizados testes onde obteve-se medições de resultados ao fim de uma aplicação de cada CNOT contida no circuito. A ideia é mostrar que com o aumento gradativo do uso da CNOT o erro aumentava junto até chegar em uma situação que o resultado teórico esperado já estaria longe do resultado prático obtido por um processador quântico.

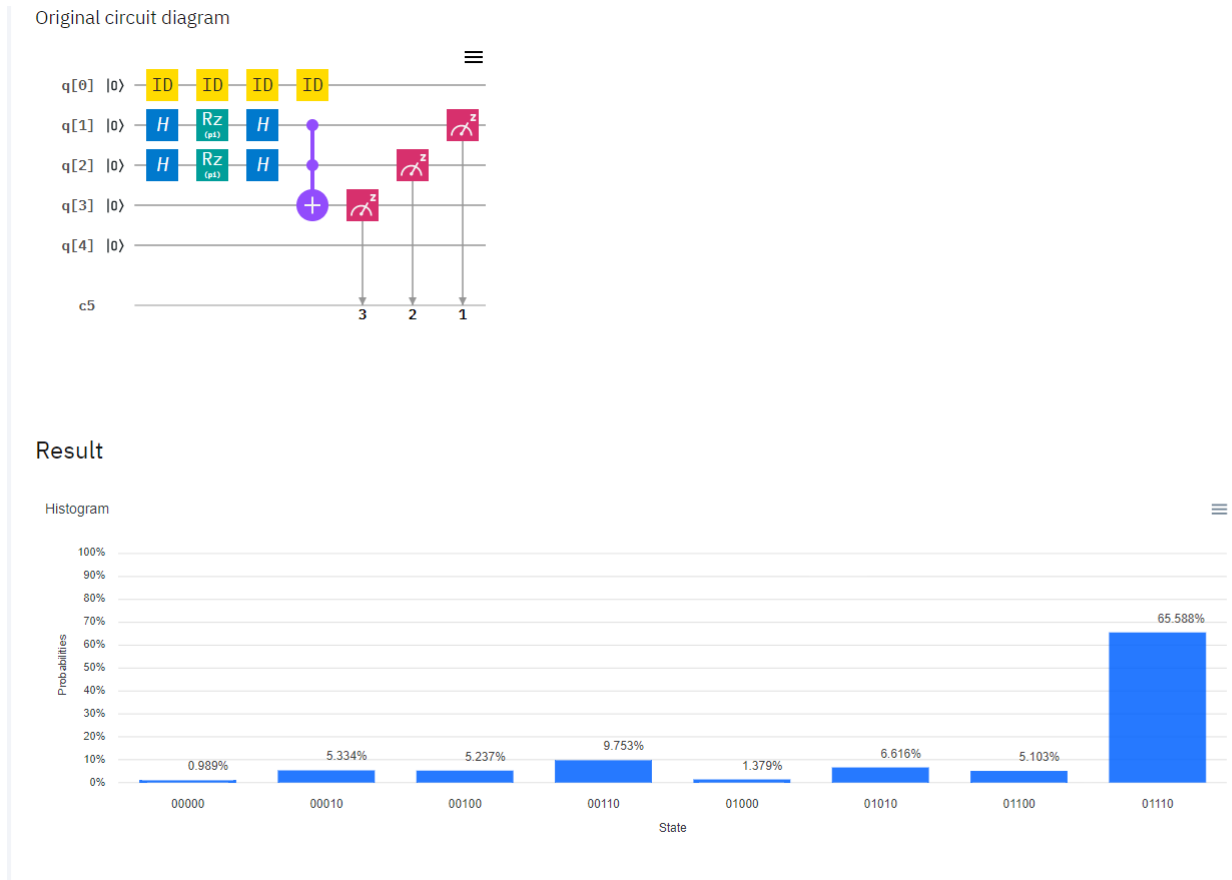
O resultado medido antes da aplicação da primeira CNOT está ilustrado na Figura 66, onde apenas os estados dos qubits 1 e 2 forma alterados para $|11\rangle$, e o resultado teórico esperado seria $|00110\rangle$.

Figura 66 – Medição do estado antes da aplicação da primeira CNOT. Observa-se que para 8192 *trials*, em aproximadamente 90% destes, o resultado seria o correto, $|00110\rangle$.



a) Teste 1: O teste 1 corresponde a primeira aplicação de uma CNOT no circuito do QPerceptron. O processador utilizado para este teste foi o IBM-VIGO, onde o qubit 1 e 2 foram os qubits de controle e o qubit 3 o qubit alvo. A Figura 67 ilustra onde foi realizado a medição. Observa-se que para 8192 *trials*, em aproximadamente 65% destes, o resultado seria o correto, $|01110\rangle$, já que os qubits 1 e 2 estariam no estado $|11\rangle$ e o resultado da CNOT seria inverter o estado do qubit 3, resultando em $|1\rangle$.

Figura 67 – Teste de erro na primeira utilização da porta CNOT no circuito do QPerceptron.

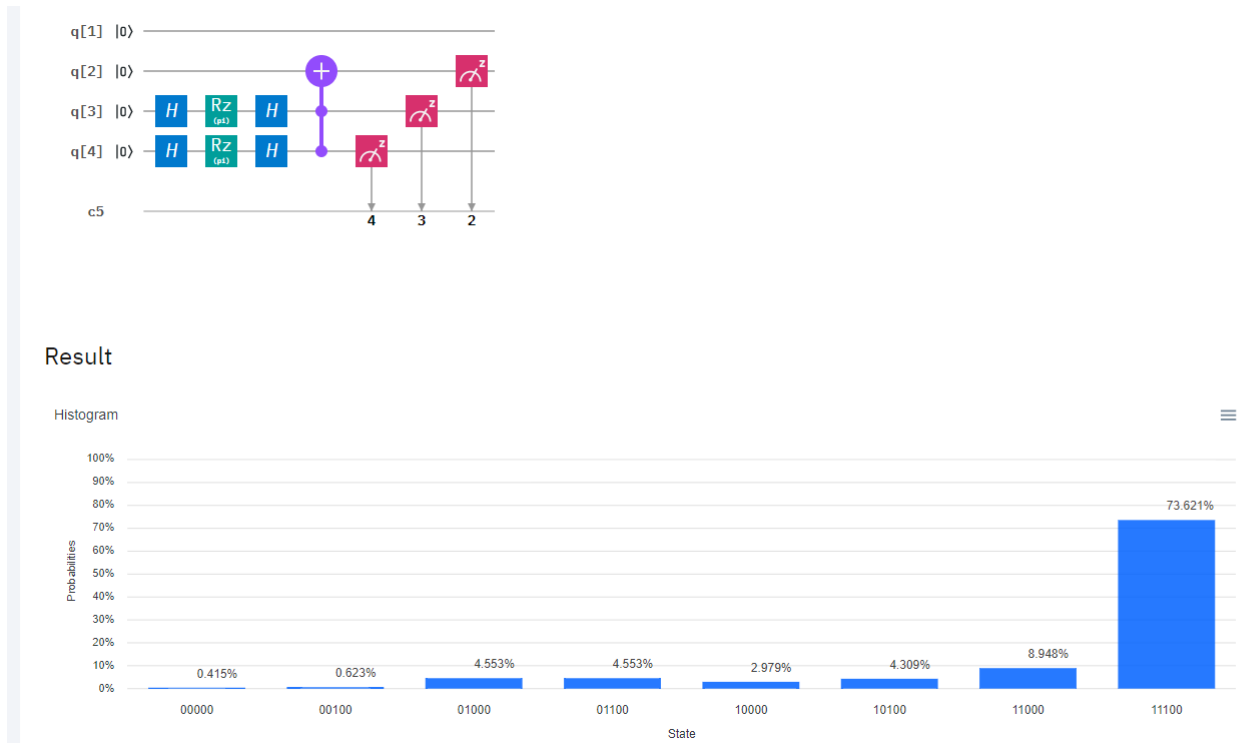


O teste 1 pode ser descrito como: o qubit 1 e 2 entrariam como $|00\rangle$ e a porta R_{ϕ} iria negá-los e assim resultar em $|11\rangle$ e com isso a porta CNOT iria ativar transformando o qubit 3 $|0\rangle$ em $|1\rangle$, os qubits 0 e 4 não se alterariam, então a resposta teórica seria $|01110\rangle$. O resultado mostra que em 8192 trials a resposta correta sairia em apenas 65.588% das vezes.

b) Teste 2: O teste 2 corresponde a primeira aplicação de uma CNOT no circuito do QPerceptron. O processador utilizado para este teste foi o IBM-VIGO, onde o qubit 3 e 4 foram os qubits de controle e o qubit 2 o qubit alvo. Percebe-se que foi mudado apenas a ordem dos qubits em relação ao teste 1, porém de acordo com o manual fornecido pela IBM-VIGO espera-se que o erro seja diferente, pois o erro da CNOT não é o mesmo para os qubits. A Figura 68 ilustra onde foi realizado a medição. Observa-se que para 8192 trials, em aproximadamente 74% destes, o resultado seria o correto, $|11100\rangle$, já que os qubits 3 e 4 estariam no estado $|11\rangle$ e o resultado da CNOT seria inverter o estado do qubit 2, resultando em $|1\rangle$.

O teste 2 pode ser descrito como: o qubit 3 e 4 entrariam como $|00\rangle$ e a porta R_{ϕ} iria negá-los e assim resultar em $|11\rangle$ e com isso a porta CNOT iria ativar transformando o qubit 2 $|0\rangle$ em $|1\rangle$, os qubits 0 e 1 não se alterariam, então a resposta teórica seria $|01110\rangle$.

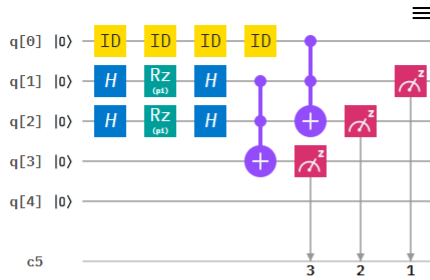
Figura 68 – Teste de erro na primeira utilização da porta CNOT no circuito do QPerceptron, com uma configuração diferente de qubits.



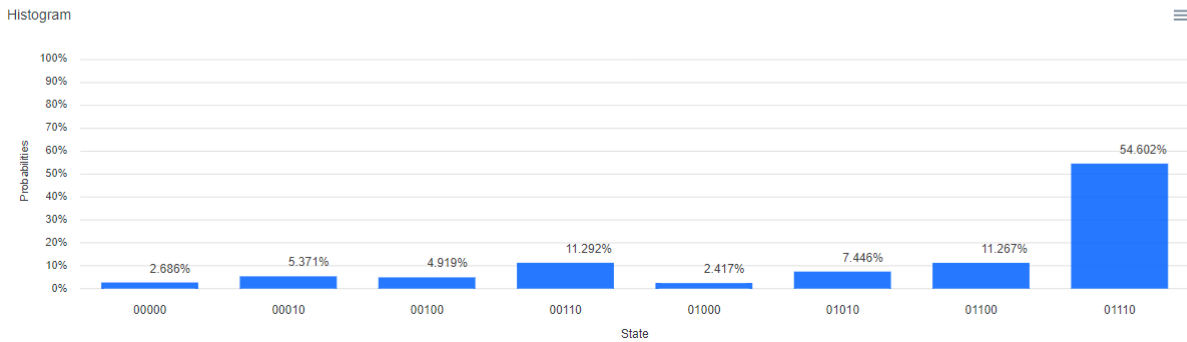
O resultado mostra que em 8192 trials a resposta correta sairia em apenas 73.621% das vezes. Percebe-se que a mudança dos qubits de controle e alvo alteraram a probabilidade de medição, isto está de acordo com que a IBM Q informa no seu manual do processador VIGO.

c) Teste 3: O teste 3 corresponde a segunda aplicação de uma CNOT no circuito do QPerceptron. O processador utilizado para este teste foi o IBM-VIGO, onde o qubit 1 e 2 foram os qubits de controle e o qubit 3 o qubit alvo para a primeira CNOT, já na segunda CNOT, os qubits 0 e 1 foram o de controle e o 2 o alvo. Agora a medição irá ocorrer após o uso das duas CNOTs. A Figura 69 ilustra onde foi realizado a medição. Observa-se que para 8192 trials, em aproximadamente 54% destes, o resultado seria o correto, $|01110\rangle$, já que os qubits 0 e 1 estariam no estado $|01\rangle$ e o resultado da segunda CNOT seria de não inverter o estado do qubit 2, resultando em $|1\rangle$.

Figura 69 – Teste de erro na segunda utilização da porta CNOT no circuito do QPerceptron.



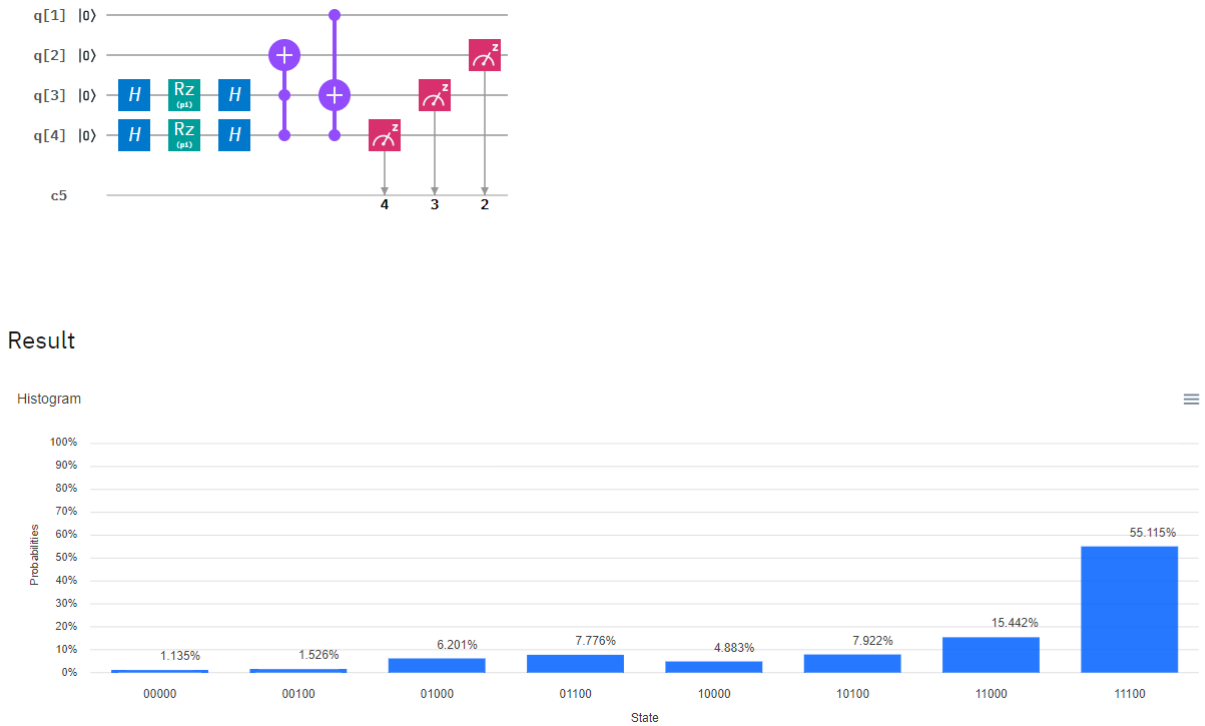
Result



O teste 3 pode ser descrito como: o qubit 1 e 2 entrariam como $|00\rangle$ e a porta R_{phi} iria negá-los e assim resultar em $|11\rangle$ e com isso a porta CNOT iria ativar transformando o qubit 3 $|0\rangle$ em $|1\rangle$, os qubits 0 e 4 não se alterariam, depois os qubits 0 e 1 seriam os qubits de controle da CNOT, estes são $|01\rangle$, ou seja, a CNOT não irá se ativar e portanto o qubit 2 permanecerá inalterado. Então a resposta teórica seria $|01110\rangle$, ou seja o mesmo resultado do teste 1, porém teoricamente a probabilidade obtida no teste 1 seria a mesma do teste 3, infelizmente o resultado mostra que em 8192 trials a resposta correta sairia em apenas 54.602% das vezes. Portanto, o uso progressivo da porta CNOT acarreta em um acúmulo de erro no resultado de medição.

b) Teste 4: O teste 4 corresponde a segunda aplicação de uma CNOT no circuito do QPerceptron. O processador utilizado para este teste foi o IBM-VIGO, onde o qubit 3 e 4 foram os qubits de controle e o qubit 2 o qubit alvo, já na segunda CNOT, os qubits 1 e 4 seriam o de controle e o 3 o alvo. Percebe-se que foi mudado apenas a ordem dos qubits, sendo esta ordem igual ao do teste 2, porém com outra aplicação de uma CNOT. A Figura 70 ilustra onde foi realizado a medição. Observa-se que para 8192 trials, em aproximadamente 55% destes, o resultado seria o correto, $|11100\rangle$, já que os qubits 1 e 0 estariam no estado $|01\rangle$ e o resultado da CNOT seria de não inverter o estado do qubit 3, resultando em $|1\rangle$.

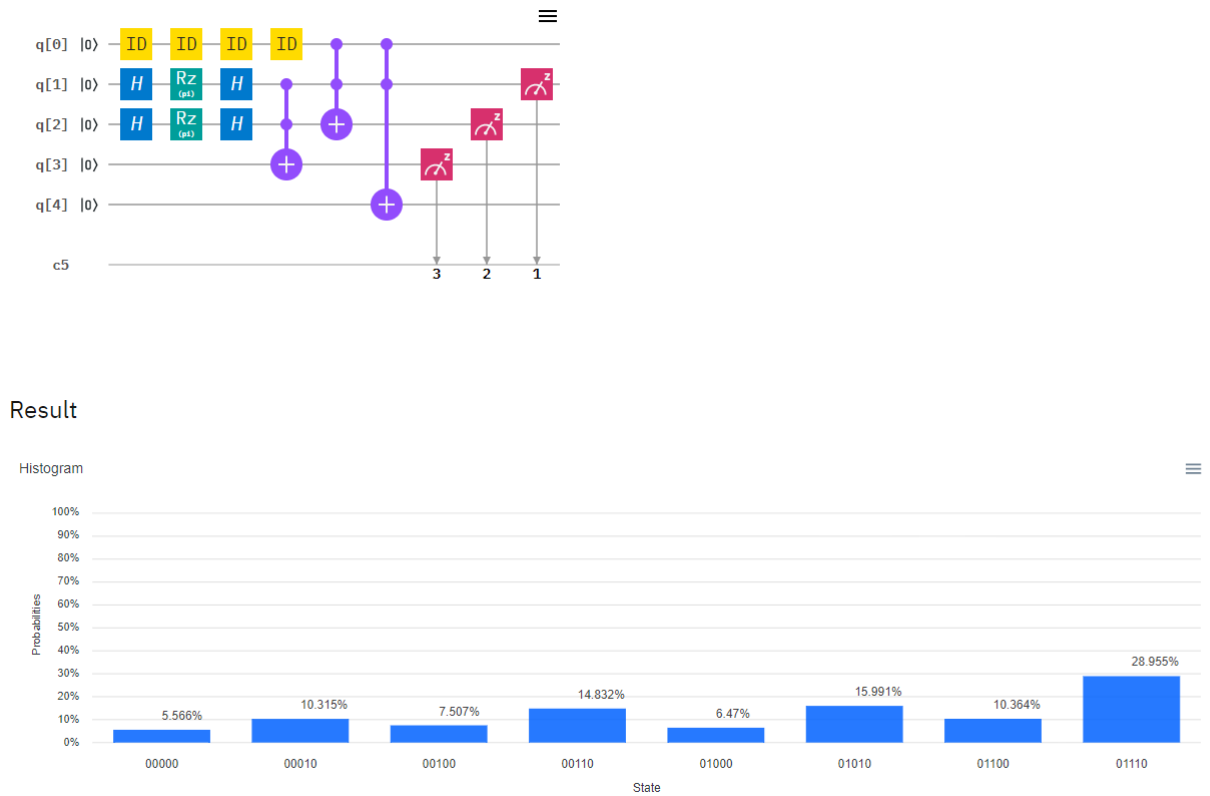
Figura 70 – Teste de erro na segunda utilização da porta CNOT no circuito do QPerceptron, com uma configuração diferente de qubits.



Conforme o esperado do teste 3, o uso da CNOT iria acumular o erro, porém utilizando outra rearanjo de qubits, o resultado mostra que em 8192 trials a resposta correta sairia em apenas 55.115% das vezes, um aumento nem tanto expressivo em relação ao teste 3.

c) Teste 5: O teste 5 corresponde a segunda aplicação de uma CNOT no circuito do QPerceptron. O processador utilizado para este teste foi o IBM-VIGO, onde o qubit 1 e 2 foram os qubits de controle e o qubit 3 o qubit alvo para a primeira CNOT, já na segunda CNOT, os qubits 0 e 1 foram o de controle e o 2 o alvo, no terceiro uso da CNOT os qubits 0 e 1 também seriam os de controle, mas o qubit 4 seria o alvo. Agora a medição irá ocorrer após o uso das três CNOTs. A Figura 71 ilustra onde foi realizado a medição. Observa-se que para 8192 *trials*, em aproximadamente 28% destes, o resultado seria o correto, $|01110\rangle$, já que os qubits 1 e 0 estariam no estado $|01\rangle$ e o resultado da CNOT seria de não inverter o estado do qubit 4, resultando em $|0\rangle$.

Figura 71 – Teste de erro na terceira utilização da porta CNOT no circuito do QPerceptron.



Portanto, apenas reforçando a ideia que o erro aumenta gradualmente, o resultado mostra que em 8192 trials a resposta correta sairia em apenas 28,955% das vezes, ou seja, o erro gerado do uso gradativo da CNOT acaba por piorar o resultado teórico esperado, o que explica o erro gigante obtido ao final do circuito do QPerceptron.

4 Conclusão

Ao longo deste trabalho foi discutido sobre o que é computação quântica, suas vantagens, perspectivas e aplicações. Também foi apresentado de forma breve a noção de aprendizagem de máquina em geral, além de abordar o que são redes neurais e como elas funcionam. A junção destes dois tópicos culminaram na aprendizagem de máquina quântica, e a partir deste conceito o trabalho foi embasado.

A ideia principal era mostrar uma adaptação de um algoritmo clássico para uma nova versão deste sendo fundamentada nos conceitos quânticos, onde este fosse capaz de demonstrar as vantagens da utilização de um computador quântico frente aos computadores atuais.

Além disso, o algoritmo quântico também tinha um contexto pelo qual foi proposto. A proposta era solucionar as limitações do seu antecessor clássico e melhorar a performance de velocidade do algoritmo de aprendizagem, e como evidenciado nos resultados obtidos no experimento a proposta foi alcançada, com uma certa observação. A observação a ser feita refere-se ao que foi discutido no final do experimento 2, mas deve ser comentada, pois os processadores quânticos utilizados não foram projetados sem decoerência. Apesar de que isso não seja capaz de tirar o mérito deles, visto que já é um grande avanço poder contar com estes de forma gratuita e facilmente acessível, além disso os próprios processadores quânticos da IBM informam que há um erro associado no uso da porta CNOT e como foi demonstrado em testes, estes erros acabam se acumulando durante o circuito e resultam em resultados não esperados com a teoria.

Ademais, a ideia do trabalho era propor um novo algoritmo de Perceptron quântico que diferentemente do que foi proposto por outros pesquisadores da área de aprendizagem de máquina quântica, conseguisse provar sua engenhosidade no campo teórico e prático.

Portanto, para todos aqueles que leram este trabalho espera-se que fique claro uma mensagem: a mecânica quântica e a computação quântica são áreas que merecem uma atenção fundamental, uma vez que conceitos errôneos sobre os mesmos estão presentes na visão de muitas pessoas. É preciso pesquisar e estudar sobre estes assuntos antes de formular ideias a respeito de sua aplicabilidade, pois como mostrado ao longo deste trabalho há muitas e muitas aplicações para a mecânica quântica, computação quântica e informação quântica e que elas conseguem teoricamente demonstrar um avanço enorme na resolução de problemas que até então pareciam ser difíceis de serem solucionados. Conclui-se que este trabalho é importante ressaltar que no futuro estas áreas vão ganhar um grande e fundamental papel para o desenvolvimento de novas tecnologias, e a aprendizagem de máquina quântica é apenas uma de muitas de suas aplicabilidades.

A Apêndice

A.1 Algoritmos quânticos

Uma das promessas dos computadores quânticos é habilitar novos algoritmos que viabilizam solucionar problemas que exigem recursos exorbitantes para sua solução em um computador clássico. Um algoritmo quântico é um conjunto de instruções para um computador quântico, esses em sua maioria envolvem uma série de transformações no estado quântico seguidas por uma medição. Atualmente, pouquíssimos algoritmos quânticos são conhecidos, porém existem duas grandes classes de algoritmos quânticos conhecidas: A primeira classe de algoritmos é baseada na transformada de Fourier quântica de Shor, a segunda classe de algoritmos é baseada no algoritmo de Grover para realizar a pesquisa quântica.

A.1.1 Algoritmo de Grover

Um tipo de classe de algoritmos muito conhecido, principalmente na área da ciência da computação, é a classe de algoritmos de busca. O seguinte exemplo ilustra um problema onde a aplicação deste tipo de algoritmo é utilizada: "Seja uma coleção de elementos, em que cada elemento possui um identificador único. Devemos encontrar o elemento da coleção que possui o mesmo identificador de uma dada chave de busca ou verificar que não existe nenhum elemento na coleção com a chave dada.", resumidamente o problema da busca (ou pesquisa) pode ser definido como "Dado um conjunto de elementos, onde cada um é identificado por uma chave, o objetivo da busca é localizar, nesse conjunto, o elemento que corresponde a uma chave específica."

Na computação quântica existe um algoritmo de busca que foi descoberto por Lov Grover, chamado de algoritmo de busca quântico ou algoritmo de Grover. O algoritmo de busca quântico resolve o seguinte problema: Dado um espaço de pesquisa de tamanho N e nenhum conhecimento prévio sobre a estrutura das informações contidas nele, queremos encontrar um elemento desse espaço de pesquisa que satisfaça uma propriedade conhecida (CITE CHUANG), ou seja, bem parecido com o problema resolvido pelos algoritmos de busca clássicos. Classicamente, esse problema requer aproximadamente N operações para achar a solução, mas o algoritmo de busca quântico permite que ele seja resolvido usando aproximadamente \sqrt{N} operações.

O algoritmo de Grover fornece uma aceleração quadrática sobre os melhores algoritmos clássicos possíveis. Esse algoritmo tem muitas aplicações em potencial, das quais apenas algumas são ilustradas: Ele pode ser usado para extrair estatísticas, como o

elemento mínimo, de um conjunto de dados não ordenados, mais rapidamente do que é possível em um computador clássico. Ele pode ser usado para acelerar algoritmos para alguns problemas no NP - especificamente aqueles problemas para os quais uma pesquisa direta por uma solução é o melhor algoritmo conhecido. Por fim, ele pode ser usado para acelerar a pesquisa de chaves para sistemas de criptografia, como o amplamente utilizado *Data Encryption Standard (DES)*.

A.1.2 Algoritmo de transformada de Fourier quântica de Shor

"Apesar da grande dificuldade de construir um computador quântico verdadeiramente de propósito geral, pode ser relativamente fácil construir uma máquina de fatoração quântica de propósito especial que possa ser usada para quebra de código. A história tem uma tendência a se repetir; não foram os primeiros computadores digitais usados para quebra de código ?" S. Y. Yan.

O algoritmo quântico para fatoração de inteiros em tempo polinomial publicado por Peter W. Shor em 1994 (LI et al., 2001) é considerado o primeiro algoritmo quântico que combina relevância prática e eficiência.

A transformada quântica de Fourier de Shor é um algoritmo quântico eficiente para realizar uma transformação de Fourier de amplitudes mecânicas quânticas. Ela não acelera a tarefa clássica de calcular transformações de Fourier de dados clássicos, mas uma tarefa importante que ela habilita é a estimativa de fase e a aproximação dos valores próprios de um operador unitário sob certas circunstâncias. Isso permite resolver vários outros problemas interessantes, incluindo o problema de localização de pedidos e o problema de fatoração. A estimativa de fase também pode ser combinada com o algoritmo de busca quântica para resolver o problema de contar soluções para um problema de pesquisa.

Referências

- ABRAHAM, H. et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2019. Citado 2 vezes nas páginas 48 e 64.
- ADAMI, C.; DOWLING, J. P. Quantum computation—the ultimate frontier. *arXiv preprint quant-ph/0202039*, 2002. Citado na página 14.
- ADAMS, D. et al. *O guia do mochileiro das galáxias*. [S.l.]: Sextante, 2004. Citado na página 12.
- ALTAISKY, M. Quantum neural network. *arXiv preprint quant-ph/0107012*, 2001. Citado 2 vezes nas páginas 39 e 63.
- BENENTI, G.; CASATI, G.; STRINI, G. *Principles of Quantum Computation and Information-Volume I: Basic Concepts*. [S.l.]: World scientific, 2004. Citado na página 24.
- BIAMONTE, J. et al. Quantum machine learning. *Nature*, Nature Publishing Group, v. 549, n. 7671, p. 195, 2017. Citado na página 36.
- BRAGA, A. de P.; FERREIRA, A. C. P. de L.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: LTC Editora Rio de Janeiro, Brazil., 2007. Citado 2 vezes nas páginas 32 e 35.
- CILIBERTO, C. et al. Quantum machine learning: a classical perspective. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society Publishing, v. 474, n. 2209, p. 20170551, 2018. Citado 2 vezes nas páginas 37 e 38.
- CROSS, A. The ibm q experience and qiskit open-source quantum computing software. In: *APS Meeting Abstracts*. [S.l.: s.n.], 2018. Citado na página 64.
- DUNJKO, V.; BRIEGEL, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, IOP Publishing, v. 81, n. 7, p. 074001, 2018. Citado 2 vezes nas páginas 36 e 37.
- FEI, L.; BAOYU, Z. A study of quantum neural networks. In: IEEE. *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*. [S.l.], 2003. v. 1, p. 539–542. Citado 2 vezes nas páginas 40 e 63.
- FEYNMAN, R. P. Simulating physics with computers. *International journal of theoretical physics*, Springer, v. 21, n. 6, p. 467–488, 1982. Citado 2 vezes nas páginas 15 e 16.
- FURSMAN, A. Z. A. Innovating with quantum computing enterprise experimentation provides view into future of computing. Accenture, 2007. Citado na página 17.
- GÉRON, A. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. [S.l.]: "O'Reilly Media, Inc.", 2017. Citado na página 31.
- JAEGER, L. *The Second Quantum Revolution: From Entanglement to Quantum Computing and Other Super-Technologies*. [S.l.]: Springer, 2018. ISBN 9783319988245. Citado 6 vezes nas páginas 14, 15, 16, 17, 18 e 19.

- JOHNSTON, N. H. E.; GIMENO-SEGOVIA, M. *Programming Quantum Computers*. [S.l.]: "O'Reilly Media, Inc.", 2019. Citado na página 47.
- KAK, S. C. Quantum neural computing. In: *Advances in imaging and electron physics*. [S.l.]: Elsevier, 1995. v. 94, p. 259–313. Citado 2 vezes nas páginas 39 e 63.
- KAYE, P. et al. *An introduction to quantum computing*. [S.l.]: Oxford University Press, 2007. Citado na página 19.
- LI, S.-S. et al. Quantum computing. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 98, n. 21, p. 11847–11848, 2001. Citado na página 81.
- LOCEFF, M. A course in quantum computing (for the community college). *Foothill College*, 2015. Citado na página 21.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 31.
- MOTTA, V. S.; CARVALHO, L. M.; MACULAN, N. Esfera de bloch: algumas propriedades. *São Paulo*, 2005. Citado na página 20.
- NIELSEN, M. A.; CHUANG, I. *Quantum computation and quantum information*. [S.l.]: AAPT, 2002. Citado 3 vezes nas páginas 16, 19 e 21.
- PARSONS, S. Introduction to machine learning by ethem alpaydin, mit press, 0-262-01211-1, 400 pp. *The Knowledge Engineering Review*, Cambridge University Press, v. 20, n. 4, p. 432–433, 2005. Citado na página 29.
- RIEFFEL, E.; POLAK, W. An introduction to quantum computing for non-physicists. *ACM Computing Surveys (CSUR)*, ACM, v. 32, n. 3, p. 300–335, 2000. Citado 2 vezes nas páginas 19 e 72.
- RIEFFEL, E. G.; POLAK, W. H. *Quantum computing: A gentle introduction*. [S.l.]: MIT Press, 2011. Citado 2 vezes nas páginas 15 e 19.
- ROJAS, R. *Neural networks: a systematic introduction*. [S.l.]: Springer Science & Business Media, 2013. Citado na página 35.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado 2 vezes nas páginas 31 e 35.
- SAGHEER, A.; ZIDAN, M. Autonomous quantum perceptron neural network. *arXiv preprint arXiv:1312.4149*, 2013. Citado 2 vezes nas páginas 39 e 63.
- SANTOS, A. C. O computador quântico da ibm e o ibm quantum experience. *Caderno Brasileiro de Ensino de Física*, v. 39, n. 1, 2017. Citado na página 64.
- SHAFEE, F. Neural networks with quantum gated nodes. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 20, n. 4, p. 429–437, 2007. Citado na página 38.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding machine learning: From theory to algorithms*. [S.l.]: Cambridge university press, 2014. Citado 2 vezes nas páginas 29 e 30.

SILVA, I. d.; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais para engenharia e ciências aplicadas. *São Paulo: Artliber*, v. 23, n. 5, p. 33–111, 2010. Citado 3 vezes nas páginas [31](#), [33](#) e [35](#).

TAMURA, K.; SHIKANO, Y. Quantum random numbers generated by the cloud superconducting quantum computer. *arXiv preprint arXiv:1906.04410*, 2019. Citado na página [45](#).

VAHID, F. *Sistemas Digitais*. [S.l.]: Bookman Editora, 2009. Citado na página [15](#).

WILLIAMS, C. P. *Explorations in quantum computing*. [S.l.]: Springer Science & Business Media, 2010. Citado 5 vezes nas páginas [23](#), [24](#), [26](#), [27](#) e [28](#).

YANLING, Z.; BIMIN, D.; ZHANRONG, W. Analysis and study of perceptron to solve xor problem. In: IEEE. *The 2nd International Workshop on Autonomous Decentralized System, 2002*. [S.l.], 2002. p. 168–173. Citado na página [38](#).