Pós-Graduação em Ciência da Computação

Fernando Maciano de Paula Neto

Advances in Quantum Neural Networks

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2018

Fernando Maciano de Paula Neto

Advances in Quantum Neural Networks

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Inteligência computacional
Orientadora: Dra. Teresa Bernarda Ludermir
Coorientador: Dr. Wilson Rosa de Oliveira

Recife
2018

Catalogação na fonte
Bibliotecária Elaine Freitas CRB 4-1790

P324a Paula Neto, Fernando Maciano de
Advances in quantum neural networks/ Fernando Maciano de Paula Neto – 2018.
135 f.: fig., tab.

Orientadora: Teresa Bernarda Ludermir.
Tese (Doutorado) – Universidade Federal de Pernambuco. Cin. Ciência da Computação. Recife, 2018.
Inclui referências apêndices.

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

006.3 CDD (22. ed.) UFPE-MEI 2018-143

Fernando Maciano de Paula Neto

Advances in Quantum Neural Networks

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 23/11/2018.

_____

Orientadora: Profa. Dra. Teresa Bernarda Ludermir

BANCA EXAMINADORA

_____

Prof. Dr. Antônio Murilo Santos Macêdo
Departamento de Física / UFPE

_____

Prof. Dr. Regivan Hugo Nunes Santiago
Departamento de Informática e Matemática Aplicada / UFRN

_____

Prof. Dr. Adenilton José da Silva
Departamento de Computação / UFRPE

_____

Prof. Dr. Tiago Alessandro Espinola Ferreira
Departamento de Estatística e Informática / UFRPE

_____

Prof. Dr. Borko Stosic
Departamento de Estatística e Informática / UFRPE

Dedico a todos os seres que vivem.

AGRADECIMENTOS

Agradeço aos meus orientadores, professora Teresa Ludermir e professor Wilson de Oliveira, cuja orientação foi essencial na realização do meu doutorado. Em nome deles, estendo minha gratidão aos professores e professoras da academia com quem tive oportunidade de conviver esses últimos anos e permitiram minha formação não só acadêmica, mas em outras esferas da vida.

Agradeço a minha querida família, minha mãe Graça, meu pai Jaime (*in memoriam*) minha irmã Marina, meu irmão Jaime, que foram alicerce de cuidados nos momentos em que precisei. A minha namorada Bruna, meu muito obrigado por todo apoio. Aos meus sobrinhos que tanto amo, João, Maria, Sarah, David. Aos meus cunhados Laurentino e Elaine. Também agradeço aos amigos e amigas, em todas as direções e tempo.

Agradeço aos professores amorosos com que tive oportunidade de aprender a cultivar o que há de melhor na minha essência, a ver a vida ampla e com sabedoria. Ao professor Lama Padma Samten e a todos os professores do passado que dedicam sua vida na preservação de uma sabedoria viva, minha máxima gratidão.

RESUMO

Redes Neurais Artificiais (RNA) têm sido utilizadas como modelos computacionais que aprendem a partir de um conjunto de dados e são capazes de extrapolar esse conhecimento a partir das generalizações inerentes ao seu processo de decisão. Com o crescimento da computação quântica como novo paradigma de processamento de informação, modelos quânticos de redes neurais têm sido propostos para coadunar os benefícios da computação quântica com os benefícios das RNAs. Os modelos quânticos de RNA existentes assumem a dificuldade de implementar a não-linearidade intrínseca dos neurônios que compõem a RNA, uma vez que tradicionalmente a computação quântica possui apenas operadores unitários. Há algumas propostas na literatura de modelos de neurônios que simulam essa não-linearidade, mas elas aparecem simulando alguma função não-linear específica, como a função de limiar ou arco-tangente. Há ainda as RNAs que possuem comportamento de memória associativa, fazendo a recuperação de informação a partir de uma entrada igual ou parecida com seus padrões armazenados. A implementação desses modelos envolve duas etapas, o processo de armazenamento e a recuperação de informação. Os modelos quânticos de memória associativa têm utilizado a superposição quântica para armazenamento e alguns modelos foram propostos para recuperação de informação. Nesse trabalho estendemos o funcionamento não-linear do Perceptron, permitindo que um neurônio quântico execute qualquer função não-linear discreta. O modelo proposto permite que o neurônio possa simular o comportamento dos neurônios clássicos assim como utilizar dos recursos intrínsecos da computação quântica como superposição e emaranhamento. Há também a proposição de um neurônio que possui memória interna e que pode armazenar informações de iterações anteriores a medida que ele é executado. Esse modelo permite incorporar informações espaço-temporais em seu modelo. Em termos de memórias associativas, propusemos a utilização de dois modelos de recuperação probabilística de informação, um não linear e outro linear, utilizando os algoritmos quânticos de Grover e transformada inversa de Fourier. Esse modelo de memória permite reconhecer informações próximas ou iguais ao conteúdo que está na memória e possui custo linear de operação.

Palavras-chaves: Computação quântica. Redes neurais quânticas. Neurônio quântico. Memórias associativas quânticas. Transformada de Fourier quântica. Funções não-lineares quânticas.

ABSTRACT

Artificial neural networks (ANNs) have been used as computational models that learn using a training dataset and are capable to generalise knowledge in their decision process. With the growing of quantum computing field, as new paradigm of information processing, quantum neural networks were proposed to join the benefits of quantum computing and benefits of ANNs. The quantum ANNs models have difficulty to implement the intrinsic non-linearity of neurons of ANNs since traditionally quantum operators are unitary. Some neuron proposed models in literature appear simulating some specific nonlinear function, as threshold or arctangent functions. There are RNA models which behaves as associative memories, doing recovering of information for a given input equal or similar with internal content of the memory. The implementation of these models involve two steps, the storing process and information recovering process. The quantum models of associative memories have used a quantum state in superposition to store the patterns in the memory and some models were proposed to recovery an information in the memory. In this work we extend the nonlinear operation of Perceptron, allowing that the quantum neuron executes any nonlinear discrete function. There is also the proposition of quantum neuron with internal memory being capable to save previous iterations of its execution.In terms of associative memories, we propose two models of information recovery, one nonlinear and other linear, using the algorithms of Grover and inverse of Fourier transform.

Key-words: Quantum computing. Quantum neural networks. Quantum neuron. Quantum associative memory. Quantum Fourier transform. Quantum nonlinear functions.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo de Grover
CNQ	Computador Neural Quântico
CQ	Computação Quântica
IA	Inteligência Artificial
IQ	Informação Quântica
MA	Memória Associativa
RNA	Rede Neural Artificial
TQF	Transformada Quântica de Fourier

SUMÁRIO

1 INTRODUÇÃO

O desenvolvimento de algoritmos quânticos para aplicações de inteligência artificial está em franco crescimento. Muitos algoritmos têm surgido utilizando a Computação Quântica (CQ) na resolução de problemas considerados complexos (BIAMONTE et al., 2017a). Porém, ainda não é uma tarefa simples criar algoritmos quânticos. Tal dificuldade existe devido a a forma de processar informação da CQ ser diferente do pensamento de resolução algorítmica usual (SHOR, 2003). Não se é permitido verificar informações do sistema quântico durante seu processamento sem que haja perda parcial ou total do conteúdo em execução (NIELSEN; CHUANG, 2000a), assim como é permitido que informações no sistema estejam entrelaçadas ou emaranhadas. Para a computação, o emaranhamento é considerado uma importante característica que pode acelerar o processamento de informação (JOZSA; LINDEN, 2003). Em termos de simulação, embora existam simuladores quânticos em computadores convencionais, essa simulação acontece com custo exponencial de memória (FEYNMAN, 1982) e simuladores quânticos existentes possuem poucos qubits disponíveis e um conjunto de operadores quânticos limitado para simulação (JOHANSSON; NATION; NORI, 2012; IBM..., 2018; SANTOS, 2017).

Dentro da área dita conexionista da Inteligência Artificial (IA), existe um modelo computacional que realiza tarefas não-triviais sob inspiração do comportamento do cérebro animal. Esse modelo chamado de Rede Neural Artificial (RNA) executa tarefas de processamento de informação através de uma arquitetura distribuída de nós, que possuem complexidade mais simples, denominados *neurônios*. Esses neurônios estão conectados entre si ou com sensores externos à rede e processam informações de forma majoritariamente não-linear. A força entre essas conexões são chamadas de *pesos sinápticos*, que são utilizados para armazenar o conhecimento adquirido (HAYKIN, 1999; BRAGA; CARVALHO; LUDERMIR, 2000). As Redes Neurais Artificiais Clássicas modelam hiperplanos de separação e podem ser usadas como classificadoras ou preditoras de informação. Sua capacidade de representar qualquer função não-linear e descontínua está associada à quantidade de camadas e neurônios que possui e às funções não-lineares dos neurônios que a compõem (SIEGELMANN; SONTAG, 1991). As RNA também têm sido utilizadas como Memória Associativa (MA) (HASSOUN, 1993). Essas memórias recebem uma dada entrada que tem algum grau de similaridade com o conteúdo armazenado em sua rede. O objetivo das MA é dar como saída o padrão armazenado mais próximo da entrada fornecida. Duas etapas são importantes no funcionamento das MA. A primeira etapa envolve o treinamento para que haja a memorização dos padrões. A segunda etapa envolve o processo de recuperação de uma informação, através de valores de entrada, que podem estar parciais ou com ruídos, em relação ao que foi armazenado na rede.

Os operadores quânticos são usualmente [1] unitários e por isso a simulação não-linear do

[1] A área de **sistemas quânticos abertos** investiga o uso de operadores não-unitários na computação quântica (NIEUWENHUIZEN CLAUDIA POMBO, 2014).

comportamento dos neurônios em modelos quânticos requer algum grau de criatividade para determinar a forma como essas funções vão ser executadas em um algoritmo quântico. Não há um consenso em como essas funções vão ser executadas em um sistema quântico (SCHULD; SINAYSKIY; PETRUCCIONE, 2014b). Em termos de neurônios quânticos, trabalhos foram propostos utilizando algumas funções limiar e limiar amortecidas (por exemplo, a arco-tangente). Se tratando de memórias quânticas associativas, a superposição quântica tem sido utilizada como forma de representar um conjunto exponencial de informação em um número linear de unidades computacionais (VENTURA; MARTINEZ, 2000; TRUGENBERGER, 2002a). Para o processo de recuperação, há alguns algoritmos propostos que recuperam uma dada informação parcial ou com ruídos em uma memória quântica.

Nesse trabalho, nós propomos estender os modelos quânticos de neurônios e de memórias associativas existentes, resolvendo limitações encontradas em seu funcionamento e buscando inspiração biológica para incluir comportamentos mais diversos aos modelos. Propomos a utilização do algoritmo de estimativa de fase utilizando a transformada de Fourier inversa para calcular qualquer função de ativação de um neurônio quântico. Tendo em vista os resultados dos experimentos que demonstram que unidades de processamento neuronais usando funções de ativação diferentes podem resolver com maior taxa de acerto determinadas classes de problemas (XU et al., 2015; ALMEIDA; LUDERMIR, 2010), a definição de uma arquitetura de neurônio quântico que compute qualquer função de ativação tem sido uma necessidade. Também é necessária essa definição para que algoritmos de treinamento possam ser propostos (SCHULD; SINAYSKIY; PETRUCCIONE, 2014b). Um modelo de perceptron quântico e neurônio quântico com memória interna também é proposto, uma vez que foram encontradas evidências que neurônios biológicos possuem memória interna dinâmica (AIHARA; TAKABE; TOYODA, 1990). Alguns comportamentos dinâmicos foram encontrados em iterações entre operadores quânticos, que possuem memória, com operadores lineares e não-lineares, apresentando caos e bifurcação. A presença de caos e bifurcação[2] tem sido considerados catalisadores de aprendizagem em problemas de otimização e em modelos neuronais artificiais. Em (POLLACK, 1989), resultados são encontrados apontando que uma representação de certos tipos de memórias auto-associativas pode conter características caóticas de auto-similaridade, o que implicaria no aparecimento de comportamentos caóticos. Em (YANG; LI; CHENG, 2007; NIMBARK; SUKHADIA; KOTAK, 2014), funções caóticas são utilizadas para resolver problemas de otimização de funções, implicando em diferentes comportamentos dependendo das características das funções utilizadas. Funções caóticas tem sido utilizadas para incrementar a performance durante o treinamento de uma rede neural (VERSCHURE, 1991; QUANG; KHOA; NAKAGAWA, 2007; ASADUZZAMAN et al., 2012). Nós também propomos dois modelos de recuperação de informação em memória quântica, um utilizando um operador não-unitário e outro utilizando os algoritmos quânticos de Grover e da transformada inversa quântica de Fourier.

[2] Sistemas que possuem caos, *i.e* sistemas caóticos, são sistemas que possuem sensibilidade às condições iniciais de funcionamento. Em outras palavras, possuem comportamentos que divergem exponencialmente quando mudanças na entrada, com ordem de grandeza muito pequenas, são aplicadas STROGATZ.

Esta tese possui uma compilação de artigos produzidos pelo autor que apresentam os resultados desta tese. Os artigos completos estão no Apêndice. Neste capítulo, contextualizamos o trabalho e discutimos os modelos propostos. Na Seção 1.1, apresentamos os objetivos desta tese e o fluxo de trabalho realizado para a sua execução. A Seção 1.2 trata dos conceitos mais importantes da computação quântica que são utilizados nesta tese. A Seção 1.3 trata da descrição de alguns dos principais modelos de redes neurais clássicas e quânticas. A contribuição desta tese para as redes neurais quânticas está descrita na Seção 1.4. As memórias associativas quânticas são discutidas na Seção 1.5 e as contribuições desta tese para esta área são descritas na Seção 1.6. A Seção 1.7 trata das publicações realizadas durante o período de doutoramento. A conclusão e os possíveis trabalhos futuros estão na Seção 1.9. No Apêndice deste trabalho estão as publicações ou submissões de artigos que detalham os modelos propostos desta tese.

1.1 OBJETIVOS E FLUXO DE TRABALHO

A escolha da função de ativação não-linear para as unidades de processamento de uma rede neural pode modificar a sua eficiência (ALMEIDA; LUDERMIR, 2010) e sua computabilidade (SIEGELMANN; SONTAG, 1992). Em outras palavras, quer dizer que a escolha da funções dos neurônios modifica a velocidade de treinamento, sua taxa de acerto e a classe de problemas que é capaz de resolver. É também uma questão em aberto como executar qualquer função de ativação não-linear em um neurônio em uma arquitetura quântica (SCHULD; SINAYSKIY; PETRUCCIONE, 2014a). Isso dificulta a proposição de algoritmos de aprendizado quânticos uma vez que não foi definido um modelo mais geral de neurônio que teria comportamentos equivalentes a um neurônio clássico e que possuiria as propriedades de sistemas quânticos. Há ainda evidências que alguns neurônios biológicos possuem estado-interno, ou seja, que possuem informações de seu funcionamento em instantes anteriores (AIHARA; TAKABE; TOYODA, 1990; FICHTNER; INOUE; OHYA, 2012). Tal funcionamento permite armazenar conteúdo temporal e pode ser útil em tarefas como previsão de séries temporais, reconhecimento de fala ou identificação de sistemas dinâmicos (GRAVES; MOHAMED; HINTON, 2013; MIKOLOV et al., 2010; LEE; TENG, 2000; YU, 2004; ZHANG; PATUWO; HU, 1998; HOCHREITER; SCHMIDHUBER, 1997). Construir modelos artificiais bioinspirados, considerando que tal mimese facilita a resolução de problemas complexos, tem sido área de pesquisa dentro da inteligência artificial há algum tempo (LEPORA; VERSCHURE; PRESCOTT, 2013).

Não há ainda modelos de memórias associativas quânticas unitárias que façam recuperação de informação com taxa de acerto ótimas.As funções que recuperam informações nas memórias quânticas associativas levam em conta informações da distância de *Hamming* entre a entrada usando um operador Hamiltoniano, ou recuperam informação usando o algoritmo de busca (Grover) parametrizado em função do estado-pergunta ou ainda usam operadores não-lineares para fazer a recuperação de informação.

Este trabalho tem como objetivo contribuir na proposição de modelos quânticos de redes neurais e memórias associativas, tendo em vista as limitações dos modelos existentes.

Os objetivos específicos são:

1. Propor um modelo de memória quântica associativa que considera o uso de um operador não-linear para a recuperação de informação quântica em uma memória associativa quântica;

2. Propor outro modelo de memória que possui uma recuperação de informação em uma memória associativa quântica usando o algoritmo quântico inverso de Fourier para calcular a distância entre um dado estado-pergunta e as informações armazenadas na memória, e o algoritmo quântico de Grover que maximiza a probabilidade de recuperar a informação quântica esperada. Também discutiremos

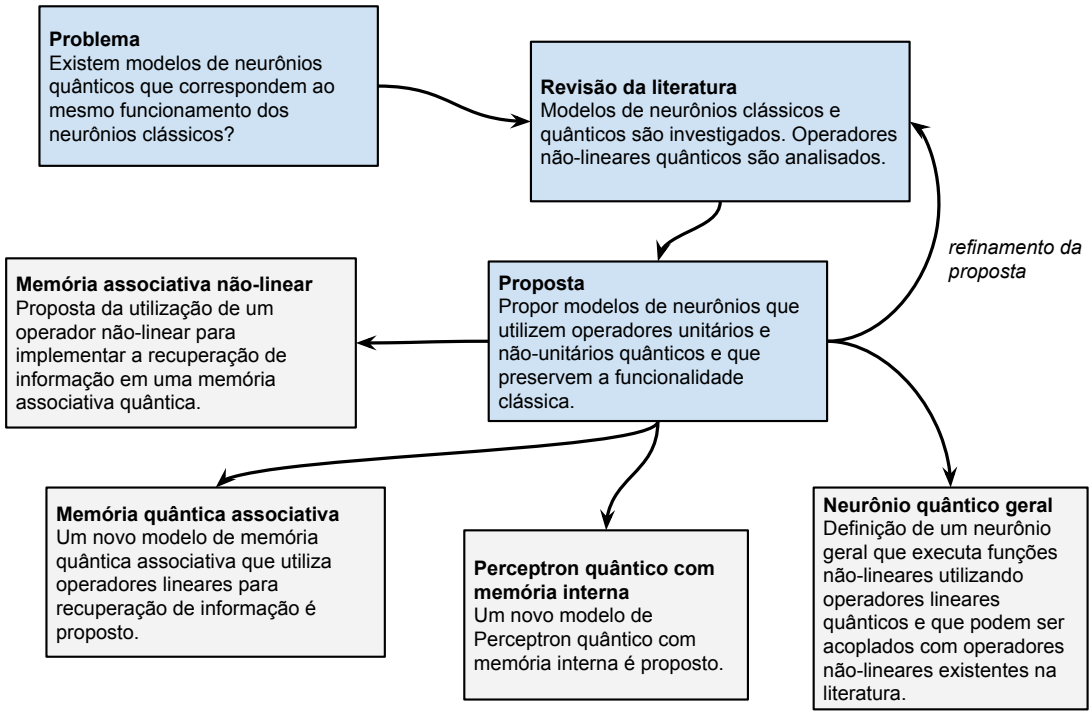
3. Propor um modelo de Perceptron quântico com memória interna;

4. Propor um modelo de neurônio quântico que executa qualquer função de ativação;

5. Propor um modelo de neurônio que executa qualquer função de ativação e que possui memória interna. Propor modelos quânticos complementares, em termos de funcionamento, aos modelos existentes de forma a expandir suas funcionalidades e simular os modelos clássicos.

Iniciamos o fluxo de execução do trabalho com objetivo de cumprir os objetivos esperados. Foi feita a análise da bibliografia dos assuntos relacionados e depois propusemos novos modelos quânticos que respondem plausivelmente aos problemas encontrados. A Figura 1 mostra este fluxo de trabalho.

Figura 1 – Fluxo de execução do trabalho.



Fonte: Feita pelo autor.

1.2 COMPUTAÇÃO QUÂNTICA E INFORMAÇÃO QUÂNTICA

Como construir algoritmos quânticos eficientes? Qual a classe de problemas em que os computadores quânticos são mais rápidos do que os computadores clássicos? Computador quântico é capaz de modelar qualquer sistema físico encontrado na natureza? Essas e outras muitas perguntas em aberto são discutidas dentro da área da CQ e Informação Quântica (IQ). Mas como essas duas áreas podem ser definidas? NIELSEN; CHUANG escrevem em seu livro:

"Por computação quântica e informação quântica entendemos o estudo das tarefas que podem ser realizadas pelo processamento de informação contida em sistemas quânticos."

Isso significa que os sistemas, ou composição de sistemas, que obedecem aos postulados da mecânica quântica são seus objetos de estudo. Pelo fato da CQ e IQ permitirem explorar sistemas quânticos utilizando uma linguagem matemática, é possível, também, utilizá-los para:

"(...)desenvolver ferramentas que agucem nossa intuição sobre mecânica quântica e tornem suas previsões mais transparentes para a mente humana (NIELSEN; CHUANG, 2000a)."

A CQ atualmente aparece como alternativa à computação clássica por dois motivos centrais: (a) devido ao processo de miniaturização dos dispositivos eletrônicos, previstos pela Lei de Moore (MOORE, 1998), efeitos quânticos passam a interferir no funcionamento de seus componentes. Modificar o paradigma de computação, incluindo a quântica, permite que interações de elementos físicos infinitamente menores possam servir como base para a computação; (b) a justificativa complementar é que a computação quântica possui algoritmos

quânticos propostos cujo custo de resolução de problemas é menor do que os algoritmos clássicos conhecidos. Além de que um computador clássico pode simular um computador quântico mas não de maneira eficiente (FEYNMAN, 1982). A construção de um computador quântico universal, proposto por DEUTSCH, permite simular sistemas quânticos eficientemente, embora não se saiba se esse modelo possa ser compatível com qualquer sistema físico arbitrário. Há algoritmos quânticos que realizam tarefas com ganho de complexidade exponencial em relação aos algoritmos clássicos existentes, como o algoritmo de fatoração de números primos que é realizado com custo polinomial (SHOR, 1994), e o algoritmo de busca em uma estrutura de dados desordenado cujo custo é sublinear, $O(\sqrt{n})$ (GROVER, 1997).

Nesta sessão, vamos dissertar sobre conceitos fundamentais da CQ para processar informação que serão utilizadas nesta tese.

1.2.1 Unidade de informação

A unidade de informação da computação quântica é o quantum-bit (qubit). O qubit é uma combinação linear de estados bidimensionais complexos. O conjunto de estados $\{|0\rangle, |1\rangle\}$ forma uma base nesse espaço bidimensional complexo e é comumente utilizado, sendo chamado de *base computacional* [3]. A combinação linear desses estados pode também ser chamada de superposição de estados. A Equação 1.1 mostra a representação de um qubit, onde α e β são números complexos que obedecem à condição de normalização $|a|^2 + |b|^2 = 1$. Podemos representar o qubit em um vetor coluna com seus valores de amplitude.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{1.1}$$

Os qubits são encontrados em fenômenos físicos validados por experimentos. Eles são modelos, por exemplo, de dois estados de um elétron,da bipolarização de um fóton e do alinhamento de um spin nuclear em um campo magnético uniforme (NIELSEN; CHUANG, 2000a).

Podemos formar uma *string* de qubits, ou ainda cadeias de qubits, utilizando o operador de produto tensorial, $\otimes$, para compô-los. Podemos representar a composição de qubits equivalentemente usando as seguintes notações $|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\rangle\,|\psi_2\rangle = |\psi_1, \psi_2\rangle =$ Na Equação

[3] A notação usada para representar os estados da base de um sistema quântico usando o formato $|.\rangle$ é atribuída a Dirac (DIRAC, 1939).

1.2, apresentamos como esta composição acontece.

$$
\begin{aligned}
|\phi\rangle &= (\alpha_1 |00...0\rangle + \alpha_2 |00...1\rangle + ... + \alpha_n |11...1\rangle) \otimes (\beta_1 |00...0\rangle + \beta_2 |00...1\rangle + ... + \beta_m |11...1\rangle) \\
&= \alpha_1(\beta_1 |00...0\rangle |00...0\rangle + \beta_2 |00...0\rangle |00...1\rangle + ... + \beta_m |00...0\rangle |11...1\rangle) \\
&+ \alpha_2(\beta_1 |00...1\rangle |00...0\rangle + \beta_2 |00...1\rangle |00...1\rangle + ... + \beta_m |00...1\rangle |11...1\rangle) \\
&+ ... + \alpha_n(\beta_1 |11...1\rangle |00...0\rangle + \beta_2 |11...1\rangle |00...1\rangle + ... + \beta_m |11...1\rangle |11...1\rangle) \\
&= \begin{matrix} |00...0\rangle \\ |00...1\rangle \\ ... \\ |11...1\rangle \end{matrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ ... \\ \alpha_n \end{pmatrix} \otimes \begin{matrix} |00...0\rangle \\ |00...1\rangle \\ ... \\ |11...1\rangle \end{matrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_m \end{pmatrix} = \begin{matrix} |00...00\rangle \\ |00...01\rangle \\ ... \\ |01...01\rangle \\ |01...10\rangle \\ |01...11\rangle \\ ... \\ |11...00\rangle \\ ... \\ |11...11\rangle \end{matrix} \begin{pmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_2 \\ ... \\ \alpha_1\beta_m \\ \alpha_2\beta_1 \\ \alpha_2\beta_2 \\ ... \\ \alpha_2\beta_m \\ ... \\ \alpha_n\beta_m \end{pmatrix}
\end{aligned}
\tag{1.2}
$$

Podemos representar os rótulos dos estados da base em um dado estado quântico utilizando seus valores decimais correspondentes. Por exemplo, o estado $|\psi\rangle = \frac{1}{\sqrt{3}} |111\rangle + \frac{\sqrt{2}}{\sqrt{3}} |101\rangle$ também pode ser representado por $|\psi\rangle = \frac{1}{\sqrt{3}} |\mathbf{7}\rangle + \frac{\sqrt{2}}{\sqrt{3}} |\mathbf{5}\rangle$. A notação $\langle .|$ representa o vetor conjugado complexo transposto do vetor $|.\rangle$. A introdução da notação usando valores decimais pode confundir o leitor em relação a quantidade de qubits utilizados em um algoritmo quântico. Por exemplo, qual seria a representação binária do estado $|\mathbf{2}\rangle$? Seria $|\mathbf{10}\rangle$, ou ainda $|\mathbf{010}\rangle$? Esta ambiguidade pode ser evitada na descrição do algoritmo, introduzindo a quantidade de qubits utilizados no procedimento.

1.2.2 Operadores

A evolução de sistemas quânticos acontece usualmente por operações unitárias que conservam a ortonormalidade dos vetores envolvidos. Esses operadores modificam o conteúdo das amplitudes complexas desses vetores, permitindo que a computação aconteça [4]. Um operador quântico

[4] Embora exista o operador quântico Identidade, $\mathbf{I}$, que não altera as amplitudes do estado envolvido. Ele pode ser utilizado na criação de operadores unitários para mais de um qubit onde não se deseja modificar o conteúdo de um ou mais qubits. Dessa forma, o operador Identidade aparece como forma de manter um operador sobre n qubits na forma matematicamente correta, ou seja, possuindo dimensão $2^n x 2^n$. Um exemplo simples para essa situação é aplicar em um sistema de 4 qubits um determinado operador U_1 no primeiro qubit, não alterar os valores de amplitude do segundo qubit, aplicar U_2 no terceiro qubit e U_3 no quarto qubit. O operador resultante nesse caso é então $U_1 \otimes \mathbf{I} \otimes U_2 \otimes U_3$.

$\mathbf{U}$, que opera sobre n qubits, tem tamanho $2^n \times 2^n$. Fixada uma base, o operador U pode ser descrito por uma matriz ($UU^\dagger = U^\dagger U = I$, onde $U^\dagger$ é transposta conjugada de U e I é matriz identidade). O estudo de operadores não-unitários na computação quântica tem sido abordado na área de estudo de sistemas quânticos abertos (BREUER; PETRUCCIONE, 2002).

Nas Equações 1.3 e 1.4 estão três bem conhecidos operadores quânticos, sua representação matricial e seu funcionamento na base computacional. O operador Identidade, **I**, é o operador que não modifica os estados envolvidos, o operador NOT, **X**, inverte o valor das amplitudes dos estados da base envolvidos e o operador Hadamard, **H**, cria uma superposição de iguais amplitudes, com a possibilidade de mudança de fase se o estado de entrada for $|1\rangle$. O operador de dois qubits Não-controlado, **CNOT**, está apresentado na Equação 1.5. Ele inverte o conteúdo do segundo qubit, chamado de qubit alvo, se o primeiro qubit, chamado de qubit de controle, for $|1\rangle$. O Operador Não-Controlado junto com operadores unitários de um único qubit são suficientes para representar qualquer porta lógica de múltiplos qubits.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{array}{l} \mathbf{I}\,|0\rangle = |0\rangle \\ \mathbf{I}\,|1\rangle = |1\rangle \end{array} \quad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{array}{l} \mathbf{X}\,|0\rangle = |1\rangle \\ \mathbf{X}\,|1\rangle = |0\rangle \end{array} \tag{1.3}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{array}{l} \mathbf{H}\,|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}\,|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle) \end{array} \tag{1.4}$$

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{array}{l} \mathbf{CNOT}\,|00\rangle = |00\rangle \\ \mathbf{CNOT}\,|01\rangle = |01\rangle \\ \mathbf{CNOT}\,|10\rangle = |11\rangle \\ \mathbf{CNOT}\,|11\rangle = |10\rangle \end{array} \tag{1.5}$$

Podemos generalizar o funcionamento do operador Não-Controlado. A ideia de que haja um qubit de controle que, se ativo, provoque uma aplicação do operador X no qubit alvo, é estendível para a aplicação de qualquer operador U. Ou seja, caso o qubit de controle seja ativo, o operador U é aplicado ao segundo qubit, o qubit alvo. O operador que realiza essa operação é chamado U-controlado.

Uma vez que um operador quântico é implementado, é possível dar como entrada mais de um valor possível ao operador, usando um estado quântico em superposição. Esse operador então pode calcular o valor de sua função para alguns ou todos valores de entrada. Essa característica da CQ é conhecida como *paralelismo quântico*.

Apesar de ser possível calcular uma função para vários valores de entrada, a informação quântica processada é recuperada (lida) classicamente, através de um operador de medição, que colapsa (lê com perdas) uma superposição de estados da base em um estado da mesma base ou de base diferente[5]. Isso significa que um estado quântico superposto será lido e um

[5] Medir em uma base diferente da base utilizada na representação de um estado superposto é possível porque podemos representar uma superposição de estados da base computacional em um estado quântico em outra base ortonormal $|a\rangle$ e $|b\rangle$ (NIELSEN; CHUANG, 2000a).

conteúdo clássico será verificado. Para um dado estado $|\psi\rangle = c_0 |\mathbf{0}\rangle + c_1 |\mathbf{1}\rangle ... c_n |\mathbf{n}\rangle$, podemos fazer apenas uma medição em $|\psi\rangle$ com probabilidade de enxergar um dado estado da base $|e\rangle$ com $|\langle\psi | e\rangle|^2$ de chance.

Por exemplo: $|\phi\rangle = \frac{1}{2}|00\rangle + \sqrt{\frac{3}{4}}|11\rangle$ for medido, a probabilidade de ver $|00\rangle$ é $p_{|00\rangle} = \langle\phi | 00\rangle = |\frac{1}{2}|^2 = 1/4$.

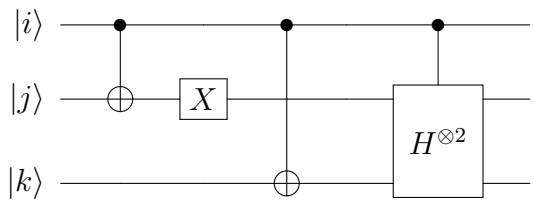
Há estados quânticos em que não podem ser representados utilizando o produto tensorial de estados mais simples. Esses estados são ditos *emaranhados*. Na Equação 1.6 estão apresentados 4 exemplos de estados emaranhados bem conhecidos na literatura da quântica como *estados de Bell*, ou ainda, *estados EPR*, em homenagem aos primeiros cientistas a estudarem as propriedades dos estados emaranhados, Einstein, Podolsky e Rosen (**??**).

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, |\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, |\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}, |\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \quad (1.6)$$

As operações quânticas, no contexto da computação quântica, podem ser representadas graficamente por *circuitos quânticos*. Esta representação considera que qubits são fios e operadores quânticos são caixas identificadas pelo rótulo de acordo com a operação que executam. O fluxo de execução (e leitura) de um circuito quântico é o mesmo do caso clássico, da esquerda para a direita.

A Figura 2 mostra um exemplo de um circuito quântico composto por um $\mathbf{CNOT}$, onde o círculo preenchido representa o qubit de controle, um operador $\mathbf{X}$ e um operador U-Controlado, onde $U = H \otimes H = H^{\otimes 2}$ [6]. Ou seja, as linhas verticais em um circuito quântico não representam propagação de informação mas um sincronismo de informação. Em outras palavras, um fio vertical indica que existe uma dependência de aplicações de operadores em função da valoração dos qubits envolvidos.

Figura 2 – Exemplo de circuito quântico com dois Não-Controlados, um operador $\mathbf{X}$ e um operador U-Controlado onde U aqui é $H \otimes H$.



Fonte: feito pelo autor.

1.2.3 Transformada Quântica de Fourier

A base teórica da computação quântica estava bem definida sobre os alicerces da mecânica quântica, mas nenhum algoritmo quântico tinha surgido até o início da década de 90 mostrando

[6] Quando o operador tensorial é aplicado sobre um mesmo operador U n vezes, é costumeiramente utilizada a notação $U^{\otimes n}$. A notação U^n indica que o operador foi n vezes multiplicado entre si.

alguma vantagem computacional em relação aos algoritmos clássicos existentes. Em 1994, Peter Shor propõe um algoritmo polinomial para fatorar números inteiros (SHOR, 1994). Isso impulsionou a área experimental e teórica da computação quântica em busca de novos algoritmos e de plataformas implementáveis da computação quântica (RIEFFEL; POLAK, 2000). Vamos explicar, nesta sessão, o algoritmo da Transformada Quântica de Fourier (TQF), que é utilizada no algoritmo de Shor, e nos trabalhos propostos desta tese.

O operador da TQF, inspirado no operador de transformada discreta de Fourier, é o operador linear aplicado a um espaço complexo N-dimensional, $|0\rangle, \cdots, |N-1\rangle$, que faz um mapeamento da informação do domínio do tempo no domínio da frequência:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle \tag{1.7}$$

Essa transformação permite armazenar uma informação de um estado quântico na fase de um novo estado completamente superposto. Aqui, temos dois exemplos de dois estados, $|001\rangle$ e $|100\rangle$, cujas informações são mapeadas, via TQF, no conteúdo da fase de um novo estado. É possível verificar que o peso associado à fase, relacionada a um dado estado da base, é proporcional ao valor desse estado k, ao valor da informação a ser armazenada j, e proporcionalmente também a $2\pi i$, onde i é a componente complexa normalizada $\sqrt{-1}$, assim como inversamente proporcional a $N = 2^n$, onde n é a quantidade de qubits na string envolvidas na transformação. Abaixo, estão dois exemplos de strings de qubits aplicadas à Transformada Quântica de Fourier:

$|\mathbf{1}\rangle = |001\rangle \xrightarrow{TQF} \frac{1}{8}(e^{2\pi i \cdot 1 \cdot 0/8} |000\rangle + e^{2\pi i \cdot 1 \cdot 1/8} |001\rangle + e^{2\pi i \cdot 1 \cdot 2/8} |010\rangle + e^{2\pi i \cdot 1 \cdot 3/8} |011\rangle + e^{2\pi i \cdot 1 \cdot 4/8} |100\rangle + e^{2\pi i \cdot 1 \cdot 5/8} |101\rangle + e^{2\pi i \cdot 1 \cdot 6/8} |110\rangle + e^{2\pi i \cdot 1 \cdot 7/8} |111\rangle)$

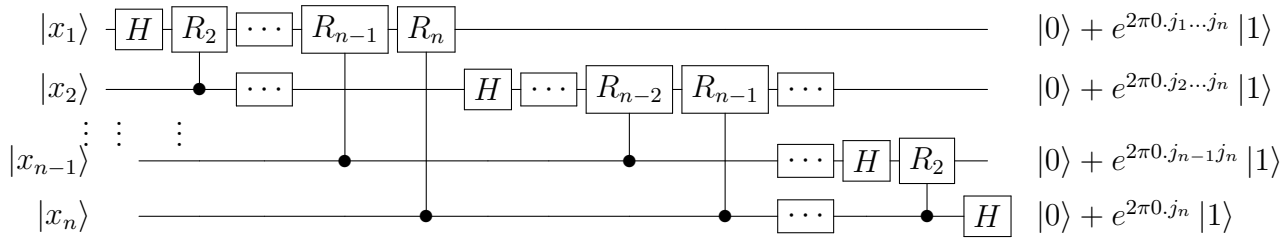
$|\mathbf{4}\rangle = |100\rangle \xrightarrow{TQF} \frac{1}{8}(e^{2\pi i \cdot 4 \cdot 0/8} |000\rangle + e^{2\pi i \cdot 4 \cdot 1/8} |001\rangle + e^{2\pi i \cdot 4 \cdot 2/8} |010\rangle + e^{2\pi i \cdot 4 \cdot 3/8} |011\rangle + e^{2\pi i \cdot 4 \cdot 4/8} |100\rangle + e^{2\pi i \cdot 4 \cdot 5/8} |101\rangle + e^{2\pi i \cdot 4 \cdot 6/8} |110\rangle + e^{2\pi i \cdot 4 \cdot 7/8} |111\rangle)$

Podemos representar a transformada realizada pela TQF usando outra notação equivalente para facilitar o desenho dessa transformada em um circuito quântico. Para uma transformação de uma string de n qubits, $|j\rangle = |j_1 j_2 ... j_n\rangle$, onde $0 \leq j \leq 2^n - 1$ e $N = 2^n$, podemos representá-la como está na Equação 1.8.

$$|j\rangle \rightarrow |j'\rangle = \frac{(|0\rangle + e^{2\pi i 0.j_n} |1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle)...(|0\rangle + e^{2\pi i 0.j_1 j_2 ... j_n} |1\rangle)}{2^{n/2}} \tag{1.8}$$

A representação $0.j_l j_{l+1} ... jm$ é equivalente a fração binária $\sum_{k=l}^{m} j_k 2^{-k}$. A definição da TQF na Equação 1.8 é chamada de representação de produto da Transformada de Fourier. A partir dessa representação, é imediato a construção do circuito mostrado na Figura 3, onde o operador R_x denota a transformação unitária descrita na Equação 1.9. Na Figura 3, foi omitido

Figura 3 – Circuito da Transformada Quântica de Fourier.



Fonte: circuito feito pelo autor, baseado no circuito proposto por NIELSEN; CHUANG.

por simplicidade o último passo do algoritmo no circuito, que inverte a ordem dos qubits.

$$R_x = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^x} \end{bmatrix} \tag{1.9}$$

1.2.4 Algoritmo de Grover

Encontrar uma informação em um vetor de tamanho n desordenado leva, no pior caso, classicamente, n passos. Um algoritmo probabilístico permite fazer essa busca em média com $n/2$ passos. Lov Grover propôs em 1996 um algoritmo quântico que realiza essa busca em $O(\sqrt{n})$ passos, onde n é a quantidade de elementos envolvidos na busca. Isso significa um ganho apenas quadrático em relação ao modelo clássico existente, mas ainda assim é um algoritmo bastante importante da área da computação quântica.

O Algoritmo de Grover (AG) funciona amplificando as amplitudes dos estados de interesse. Dado um estado quântico em superposição, o AG iterativamente inverte a fase do estado de interesse e inverte sobre a média a amplitude de todos os estados da base. Esse processo iterativo amplifica os estados de interesse e faz com que medi-los seja muito provável depois de alguns passos. No Algoritmo 1, há descrição do AG.

Algoritmo 1: Algoritmo de Grover

1 **Passo 1.** Inicie com o estado $|\mathbf{0}\rangle$
2 **Passo 2.** Aplique $H^{\otimes n}$
3 **Passo 3. repeat**
4 **Passo 4.** Aplique a operação de inversão de fase: $U_f(I \otimes H)$;
5 **Passo 5.** Aplique a inversão sobre a média: $-I + 2A$, onde I é a matriz identidade 2^n-por-2^n e A é uma matriz 2^n-by-2^n onde qualquer posição da matriz é $1/2^n$;
6 **until** $\sqrt{2^n}$ *vezes*;
7 **Passo 6.** Realize uma medida;

Aqui há um exemplo de encontrar o estado da base $|\mathbf{x_0}\rangle = |110\rangle$ no estado quântico abaixo:

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle).$$

Na primeira iteração, aplica-se o operador de inversão de fase

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle - |110\rangle + |111\rangle).$$

Depois o operador de inversão sobre a média

$$\frac{1}{2\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + 5\,|110\rangle + |111\rangle).$$

É possível verificar que depois de 3 iterações, a amplitude do estado desejado estará bem maior que as demais amplitudes e poderá ser lido com muito mais probabilidade em relação aos demais estados presentes.

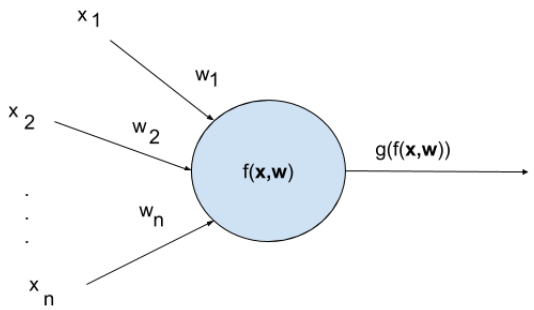
1.3 REDES NEURAIS

A área de estudo da IA é abrangente, envolvendo diversas ciências, desde a filosofia, as ciências sociais até a computacional. A datação de seu início é nebulosa. Ela está presente nos mitos e histórias antigas, dos seres criados por homens que se tornam animados e com inteligência e habilidades humanas. Está presente nas discussões antigas do que seria mente (RUSSEL; NORVIG, 2004). Porém, as pesquisas científicas em IA, como uma área da ciência capaz de propor tecnologias computacionais, começaram logo após a Segunda Guerra Mundial, por volta de 1945, da necessidade dos países bélicos estarem a frente, em termos de defesa, em relação a seus adversários. Existem vários ramos e correntes científicas na IA, que algumas vezes se conflitam ou se complementam (RUSSEL; NORVIG, 2004). Um desses ramos é chamado de conexionista, que busca inspiração neuronal biológica para criar modelos artificiais. Algumas pesquisas relacionadas à modelagem de neurônios artificiais iniciaram antes mesmo do surgimento formal da IA, sendo incorporada posteriormente em sua linha de pesquisa.

As pesquisas conexionistas iniciaram na década de 40, com um modelo de neurônio artificial proposto para representar matematicamente a operação do neurônio biológico. Neste modelo, as entradas ou estímulos externos são ponderados por informações que estão internas ao neurônio, chamadas de *pesos*. Neste modelo simples de neurônio, proposto inicialmente por MCCULLOCH; PITTS, o produto interno entre o vetor de entrada e o vetor de pesos é calculado. Caso esse cálculo ultrapasse um dado limiar, o neurônio é excitado e sua saída é um sinal positivo. Caso contrário, é dito que a saída é nula ou negativa. O Perceptron é um desses tipos de neurônios onde as entradas podem ser -1 ou 1 e os pesos podem assumir valores reais (ROSENBLATT, 1958). De forma geral, o neurônio opera sobre dois estágios. O primeiro estágio trata da interação entre as entradas e seus pesos internos. A função que produz esta interação é chamada de *função de propagação*. No segundo estágio, a *função de ativação* age no resultado da função de propagação e produz uma saída para o neurônio. Essas duas funções estão representadas na Figura 4.

Como foi dito no parágrafo anterior, a função de propagação do Perceptron é o produto interno entre a entrada e seus pesos e a função de ativação é a função degrau, mostrada na

Figura 4 – Representação gráfica do neurônio artificial. As entradas $x_1, ..., x_n$ interagem com o conteúdo interno do neurônio, seus pesos, $w_1, ...w_n$, gerando um potencial de propagação calculado pela função de propagação $f(x_1, ..., x_n, w_1, ...w_n)$. Em seguida, esse potencial é processado pela função de ativação $g(f(x_1, ..., x_n, w_1, ...w_n))$ gerando um estímulo de saída do neurônio.



Fonte: feita pelo autor

Equação 1.10.

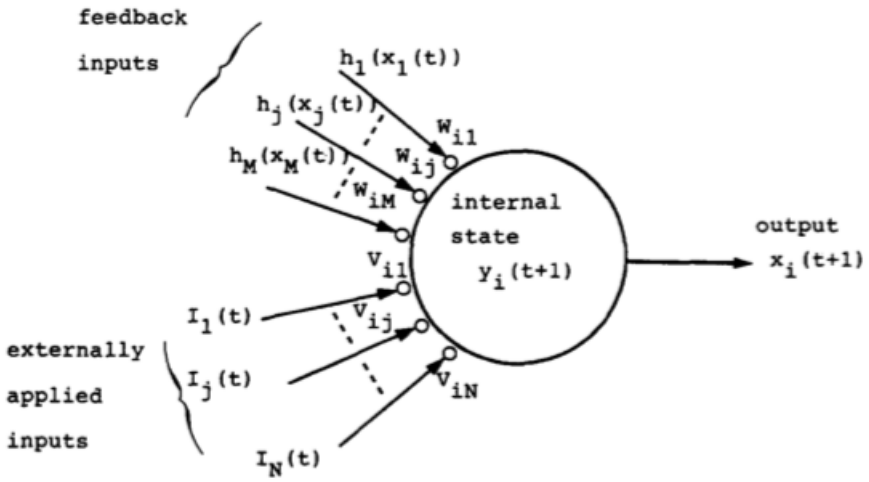
$$f(i_1, ..., i_k, w_1, ...w_k) = \begin{cases} 1, & \text{if } \sum_{k=1}^{n} w_k i_k \geq 0, \\ -1, & \text{caso contrário.} \end{cases} \tag{1.10}$$

O Perceptron, que possui função de ativação não-linear, é um classificador linear cuja convergência algorítmica no processo de treinamento é garantida. Os Perceptrons podem se conectar com outros Perceptrons, formando uma rede de várias camadas. Seus pesos podem ser ajustados por algum algoritmo de treinamento. A rede de neurônios conectados entre si forma uma RNA. Uma RNA é capaz de resolver problemas não-linearmente separáveis. Funções de ativação, tais como sigmoide logística, tangente hiperbólica são alternativas para funções de ativação, uma vez que são funções diferenciáveis e podem ser usadas em algoritmos baseados em minimização de função. Outras funções não-lineares e não totalmente diferenciáveis estão sendo usadas e tem apresentado bons resultados em problemas de classificação em redes de muitas camadas (XU et al., 2015). Existem muitas propostas de usar funções de propagação e ativação diferentes e evidências experimentais apontam que essas mudanças impactam positivamente na performance das RNAs (HAYKIN, 1999; ALMEIDA; LUDERMIR, 2010; ALMEIDA; LUDERMIR, 2008). A Equação 1.11 mostra f e g, respectivamente a função de propagação e ativação do neurônio clássico N, para um conjunto de k estímulos de entrada $x_1, ...x_k$ e seus respectivos pesos $w_1, ..., w_k$. Uma rede neural de muitas camadas com um algoritmo de treinamento é capaz de resolver problemas de aproximação de funções, tarefas de classifição, dentre outros (ZHANG, 2000).

$$N(x_1, ..., x_k, w_1, ...w_k) = g(f(x_1, ..., x_k, w_1, ..., w_k)) \tag{1.11}$$

Em 1990, AIHARA; TAKABE; TOYODA propuseram um modelo de neurônio artificial clássico com memória interna, capaz de simular comportamentos biológicos observáveis em axônios

Figura 5 – Neurônio clássico caótico proposto por Aihara *et. al.*



Fonte: AIHARA; TAKABE; TOYODA

gigantes de uma lula (MATSUMOTO et al., 1987). Deste modelo, é possível verificar também a reprodução qualitativa de caos e bifurcação durante sua dinâmica. A cada iteração, o neurônio armazena o conteúdo da iteração anterior. Esse modelo é mostrado na Figura 5. O neurônio recebe estímulos do ambiente, de outros neurônios e realimentações que são ponderados cada um por pesos diferentes.

Há evidências de que no cérebro humano há comportamentos caóticos e de bifurcação (KORN; FAURE, 2003; FREEMAN, 1992). Caos tem sido incluído como parâmetro acelerador de aprendizado das redes neurais, tanto em processos de otimização e de aprendizado (YANG; LI; CHENG, 2007; NIMBARK; SUKHADIA; KOTAK, 2014; VERSCHURE, 1991; QUANG; KHOA; NAKAGAWA, 2007), como incluindo-o na estrutura de redes neurais em função de ativação (KABIR et al., 2012; ASADUZZAMAN et al., 2012; POLLACK, 1989). A introdução do caos nesses algoritmos tem mostrado aceleração no processo de aprendizado. Da mesma forma, as redes neurais com sensibilidade às condições temporais, com realimentação, como as redes de Elman e de Jordan, tem-se mostrado hábeis em predizer comportamentos temporais e identificar sistemas dinâmicos (FERNANDEZ; PARLOS; TSAI, 1990).

1.3.1 Redes Neurais Quânticas

O conceito de Redes Neurais Quânticas foi inicialmente introduzido na década de 90 por KAK. A contribuição inicial de Kak foi ampliar a discussão dos limites de implementar uma rede neuronal clássica, considerando as pesquisas recentes sobre neurônios e redes neurais, dinâmica caótica, consciência e auto-organização, processamento de informação em microtubos do esqueleto de células e os modelos quânticos em geral:

> (...). Recent discoveries in neuroscience that cannot be placed in the reductionist models of biological information processing are examined. We learn from these stu-

> dies that **all biological systems cannot be viewed as connectionist circuits of components or systems**; there exist dynamic structures whose definition is, in part, related to the environment and interaction with other similar systems. Nevertheless, there are biological systems that are well modelled by artificial neural networks. But in general **biological systems define a concept of interdependence that is much stronger than the notion of connectionism that has been used in artificial neural networks** (KAK, 1995).

O conceito de interdependência (*interdependence*) introduzido por Kak pode ser considerado equivalente com o conceito de *correlação* trazido por NIELSEN; CHUANG. Trata-se da informação distribuída dos estados emaranhados, onde não é possível isolá-los e torná-los independentes entre si. Kak inicialmente não propõe um modelo específico de rede neural quântica ou um algoritmo de treinamento quântico. Ele introduz o conceito de Computador Neural Quântico (CNQ) (*quantum neural computer*) que ao contrário de um computador quântico genérico, que possui portas quânticas como componentes, seria uma rede neural onde processos quânticos são suportados.

Depois de Kak, diversos modelos neurais quânticos foram propostos (BIAMONTE et al., 2017a; SCHULD; SINAYSKIY; PETRUCCIONE, 2014a). A tarefa de simular o comportamento não-linear do neurônio em um computador quântico é possível ao menos na maneira como ela é executada em um computador digital, uma vez que qualquer função booleana implementável em um circuito clássico pode ser implementada polinomialmente de forma eficiente em uma máquina quântica (NIELSEN; CHUANG, 2000b). É assim que PANELLA; MARTINELLI descrevem o funcionamento de sua rede neural quântica, onde as funções Booleanas do neurônio seriam implementadas a depender do problema em que se está resolvendo. Outros modelos descrevem como vai funcionar esse mapeamento não linear entre entrada e saída do neurônio. Em termos de soluções utilizando operadores unitários, há algumas arquiteturas propostas. Há os neurônios inspirados no funcionamento da memória RAM (OLIVEIRA; SILVA; LUDERMIR, 2014; SILVA; OLIVEIRA; LUDERMIR, 2012), assim como os que simulam, em função da dinâmica de agentes físicos, como os *quantum dots*, o funcionamento de redes neurais (BEHRMAN et al., 2000). Neurônios e suas funções de ativações também foram implementadas usando algumas outras estratégias. Em (SCHULD; SINAYSKIY; PETRUCCIONE, 2015), uma função degrau é simulada usando uma função de estimativa de fase. Em (CAO; GUERRESCHI; ASPURU-GUZIK, 2017), uma função tangente periódica e função de ativação degrau é implementada utilizando um circuito probabilistico da classe *repeat-until-success (RUS) circuits* (WIEBE; KLIUCHNIKOV, 2013). A versão não-periódica desse neurônio aparece em (HU, 2018). Uma rede neural baseada em variáveis contínuas em outro modelo de computação quântica é descrita (KILLORAN et al., 2018). Alguns modelos consideram que os operadores unitários quânticos e amplitudes de estados quânticos podem ser considerados os pesos da rede neural, porém seu processo de modificação de peso envolve modificar também o operador ou a amplitude (ZHOU; DING, 2007; ALTAISKY, 2001). Essa modificação implica a possível perda da unitariedade da evolução

quântica, permitindo que esse modelo talvez não funcione em um computador quântico (SILVA; OLIVEIRA; LUDERMIR, 2015). Operadores quânticos dissipativos não-lineares também foram introduzidos para a simulação dos neurônios (GUPTA; ZIA, 2001).

Alguns experimentos também evidenciam a presença de comportamentos quânticos em estruturas biológicas e neuronais biológicas (LAMBERT et al., 2013; WEINGARTEN; DORAISWAMY; FISHER, 2016). Isso pode indicar que modelos inteligentes complexos e artificiais também poderiam explorar das características do universo quântico para otimizar sua computação.

A partir da análise dos modelos anteriores e dos resultados dos experimentos envolvendo neurônios biológicos, propusemos a discussão de um modelo neural quântico que incorporasse a possibilidade de ter comportamentos unitários e caóticos também presentes nos modelos clássicos. Há a dificuldade de se utilizar a potencialidade dos modelos das RNA com as vantagens da física e da computação quântica, como descrito por SCHULD; SINAYSKIY; PETRUCCIONE. Uma de nossas principais contribuições nesta tese é propor uma arquitetura e descrever o funcionamento do circuito que implemente um neurônio quântico que seja capaz de simular qualquer função de ativação com custo polinomial de tempo e espaço. A construção de um modelo quântico que represente um neurônio artificial não precisa acontecer da mesma forma que um modelo clássico, através de sucessivas multiplicações de matrizes de peso por um vetor de entrada. O custo de execução do modelo proposto é linear e não polinomial, como o modelo clássico. Também propomos a inclusão de uma memória interna no neurônio que armazene as entradas em tempos anteriores à execução do neurônio. Como veremos, esse tipo de memória permite ampliar as funções que esse neurônio executa, incluindo funções temporais, e apresentar comportamentos encontrados em células biológicas. Em seguida, acoplamos o neurônio a operadores quânticos não-lineares simples que representariam sua interação com o ambiente.

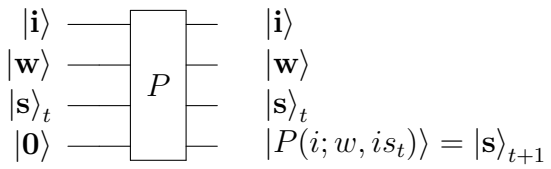
1.4 CONTRIBUIÇÕES EM REDES NEURAIS QUÂNTICAS

Propomos três modelos de Perceptron com memória interna, um modelo de neurônio que computa qualquer função de ativação não-linear e um modelo de neurônio geral, que interage com o ambiente através de operadores não-lineares quânticos, através do uso de operadores quânticos não-unitários. Iremos demonstrar através de exemplos que esses neurônios tem sensibilidade ao conteúdo da memória interna (capazes de incluir funções temporais em sua dinâmica), capazes de computar funções arbitrárias em um circuito quântico e interagir de forma caótica e não linear com operadores não-lineares conhecidos na literatura da computação quântica.

1.4.1 Perceptron quântico com memória

A partir do conceito de neurônio com memória interna, descrito por AIHARA; TAKABE; TOYODA, propomos três modelos de Perceptrons quânticos com memória interna. Esses três modelos

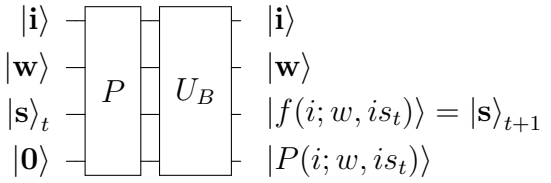
Figura 6 – Perceptron quântico com memória interna tipo A



Fonte: feita pelo autor.

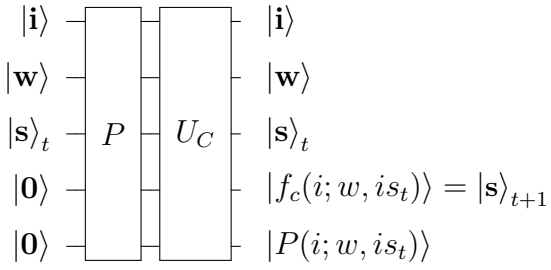
crescem em complexidade de acordo com o grau de memória disponível e a capacidade de alterar internamente sua informação. No Perceptron Quântico com memória interna tipo A, mostrado na Figura 6, o valor da saída do neurônio no tempo t é usado como estado interno na próxima iteração. Esse modelo possui pouca memória e memoriza apenas o estado anterior. O Perceptron quântico com memória interna tipo B, mostrado na Figura 7, permite que o conteúdo da memória interna seja alterado por um operador quântico reversível U_B (ou seja, um operador formado sem qubits auxiliares). O operador U_B pode ser qualquer composição de operadores unitários com variáveis $|i\rangle$, $|w\rangle$ e o estado corrente $|\mathbf{s}\rangle$. Valores de memória podem ser alterados em função do valor de entrada e pesos. O Perceptron quântico com memória interna do tipo C considera que o conteúdo interno pode ser alterado por qualquer operador quântico unitário. Ele permite que qubits auxiliares possam ser utilizados.

Figura 7 – Perceptron quântico com memória interna tipo B.



Fonte: feita pelo autor.

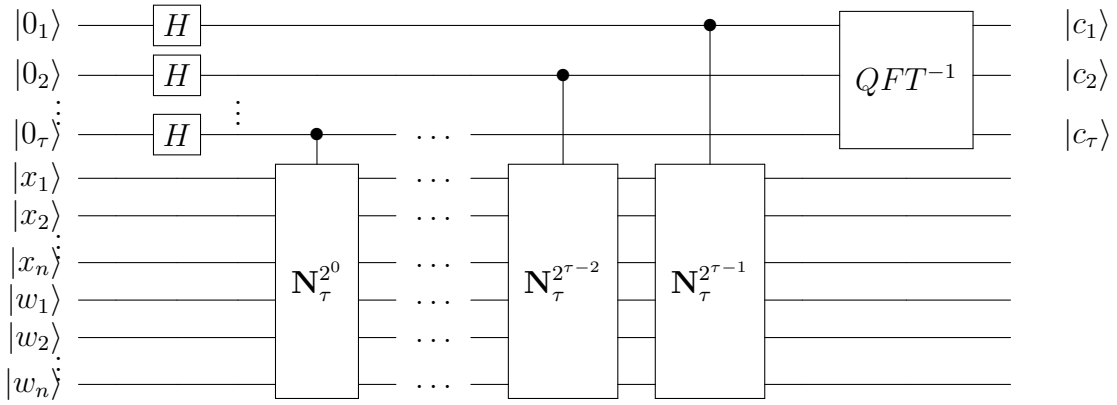
Figura 8 – Perceptron quântico com memória interna tipo C.



Fonte: feita pelo autor.

No Apêndice - Parte A desta tese, apresentamos mais exemplos de funcionamento de cada um desses modelos. Apresentamos autômatos que simulam diferentes comportamentos para

Figura 9 – Circuito do neurônio geral quântico proposto.



Fonte: feita pelo autor.

valores distintos de memórias e dependente do modelo escolhido.

1.4.2 Neurônio quântico

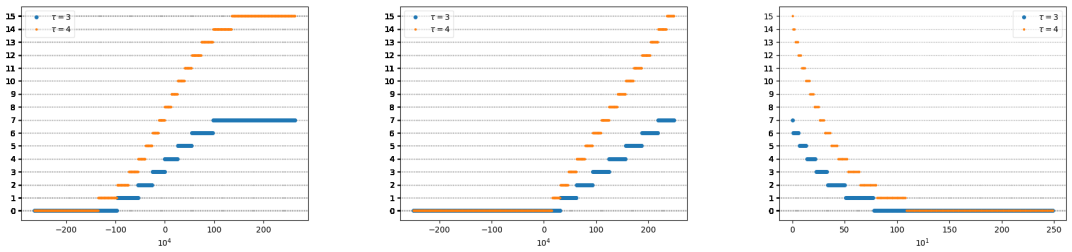
Existe a dificuldade de se implementar funções não-lineares em circuitos quânticos, de maneira otimizada, devido ao fato dos operadores quânticos serem unitários. Nós propomos um modelo de neurônio que compute funções não-lineares quaisquer. Este neurônio usa o algoritmo de estimativa de fase da transformada inversa quântica de Fourier para fazer o cálculo desta função. Sua complexidade de execução é linear, limitada pela precisão dos valores de saída do neurônio, τ. Nesta sessão, descrevemos seu funcionamento. Detalhes podem ser encontrados no artigo que está no Apêndice - Parte B deste trabalho.

O modelo proposto de neurônio que computa qualquer função não-linear está representado no circuito na Figura 9. Este neurônio armazena sua informação de saída no primeiro registrador que possui τ qubits, onde τ é a precisão do resultado de saída. Para cada um dos n estímulos de entrada no neurônio que está no registrador $|\mathbf{x}\rangle$, $|x_1, ..., x_n\rangle$, há pesos respectivos armazenados no registrador $|\mathbf{w}\rangle$, $|w_1, ..., w_n\rangle$. O neurônio computa sua função de ativação e propagação através de seu operador $\mathbf{N}_\tau$, descrito na Equação 1.12. Uma sequência de $\mathbf{N}_\tau$-Controlados permite que no primeiro registrador o valor de saída do neurônio esteja gravado na fase, em um formato que o operador inverso da transformada quântica de Fourier consiga recuperar a informação armazenada.

$$\mathbf{N}_\tau(x, w, g, f) = \sum_{x_1=0}^{2^m-1} ... \sum_{x_n=0}^{2^m-1} \sum_{w_1=0}^{2^q-1} ... \sum_{w_n=0}^{2^q-1} e^{\frac{2\pi i}{2^\tau} g(f(x_1,...,x_n,w_1,...,w_n))} |x_1, ..., x_n, w_1, ..., w_n\rangle \langle x_1, ..., x_n, w_1, ..., w_n| \tag{1.12}$$

A partir desse neurônio quântico é possível executar funções não-lineares como as exponenciais, sigmoide logísticas, de base radiais e ReLu, muito utilizadas nos trabalhos envolvendo as RNAs. Exemplos dessas funções quantizadas estão mostradas na Figura 10 para valores de

Figura 10 – Gráficos das funções (a) sigmoide, (b) ReLu e (c) exponencial para $\tau = 3$ e $\tau = 4$, que podem ser implementadas no neurônio quântico.



Fonte: feito pelo autor.

precisões τ diferentes. O custo da operação da execução deste neurônio é linear, uma vez que existe τ operadores $\mathbf{N}_\tau$, e a operação quântica inversa da transformada de Fourier é linear, para τ fixo (NIELSEN; CHUANG, 2000b). Para uma implementação do neurônio artificial em um computador quântico utilizando computação reversível, ou seja, trocando as portas lógicas por portas quânticas lógicas, o custo de operação seria quadrático, em função da quantidade de entrada e pesos envolvidos.

O neurônio quântico proposto pode simular uma Máquina de Turing clássica uma vez que executa uma função booleana. A prova de que redes neurais lógicas de uma única camada podem simular Máquinas de Turing pode ser encontrada em (OLIVEIRA; LUDERMIR, 1992). Como o neurônio quântico proposto pode ter τ qubits de saída, é suficiente um dele para fazer tal simulação.

1.4.3 Neurônio quântico com memória interna

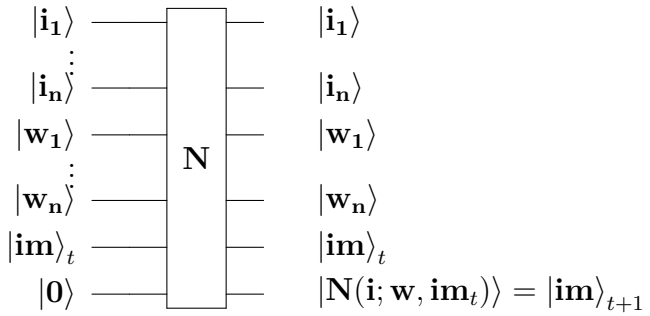
Nossa terceira contribuição envolve a proposição de um modelo de circuito para a execução de um neurônio quântico geral. Trata-se da colocação de memória interna no neurônio que executa qualquer função de ativação e da iteração desse neurônio com o ambiente.

O modelo do Perceptron quântico com memória é estendido. Os neurônio com memória interna possui também 3 tipos. O neurônio do tipo A atualiza o conteúdo da memória interna com seu valor de saída computada. O neurônio do tipo B possui um operador U_B que modifica o conteúdo de memória interna através de um conjunto de operadores sem o uso de memória auxiliar. O neurônio do tipo C possui um operador U_C que modifica a memória interna do neurônio usando qualquer função quântica, com auxílio de memória interna. Esses três tipos de neurônio estão representados nas Figuras 11, 12 e 13.

1.4.4 Neurônio quântico geral

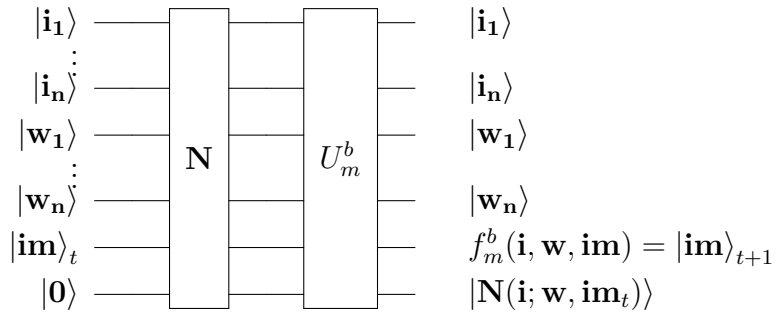
SCHULD; SINAYSKIY; PETRUCCIONE introduz a discussão de inserir sistemas quânticos abertos em modelos quânticos de neurônios, como alternativa de implementação da execução não-linear dos neurônios, uma vez que sistemas quânticos abertos permitem tal comportamento. Com o objetivo de generalizar nosso modelo de neurônio discutido na sessão anterior, propomos

Figura 11 – Neurônio Quântico com Memória Tipo A.



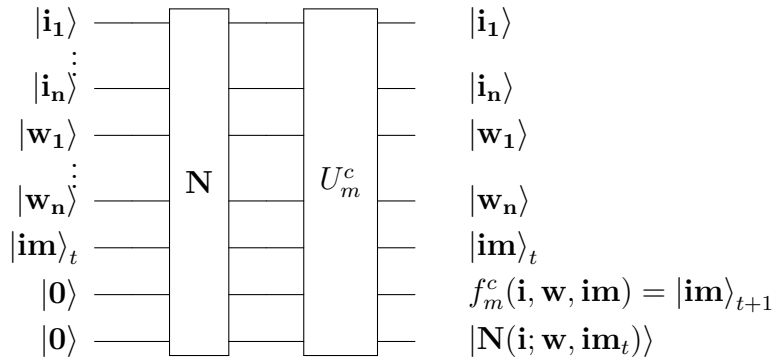
Fonte: feito pelo autor.

Figura 12 – Neurônio Quântico com Memória Tipo B.



Fonte: feito pelo autor.

Figura 13 – Neurônio Quântico com Memória Tipo C.



Fonte: feito pelo autor.

acoplá-lo ao operador não-linear que simule a interação desse neurônio com o ambiente. Esse modelo mais geral de neurônio quântico é representado na Figura 14. O operador $\mathbf{N}$ trata da execução do neurônio que executa qualquer função não-linear a partir das entradas e dos pesos, o operador U_m modifica o conteúdo interno da memória, podendo ser dos tipos A, B ou C, e operador NOp é o operador que permite a interação do neurônio com o ambiente.

Para introduzir a discussão do que esse modelo neuronal seria capaz de simular, realizamos

alguns experimentos. Tais experimentos investigam a dinâmica do neurônio no tempo utilizando operadores não-lineares conhecidos na literatura.

$$U(x,\phi) = \begin{pmatrix} cos(x) & sin(x)e^{i\phi} \\ -sin(x)e^{-i\phi} & cos(x) \end{pmatrix} \tag{1.13}$$

A dinâmica se dá pela investigação da componente de amplitude do estado da base $|0\rangle$, através de duas equações. Um delas foi proposta por KISS et al. e está na Equação 1.14. A outra forma de dinâmica, descrita na Equação 1.15, modela o comportamento incluindo a possibilidade da informação também estar na amplitude do estado $|1\rangle$ e foi nossa proposta de dinâmica advinda de testes experimentais. Nas equações, N_z é o fator de normalização. O circuito representando a dinâmica está na Figura 1.13.

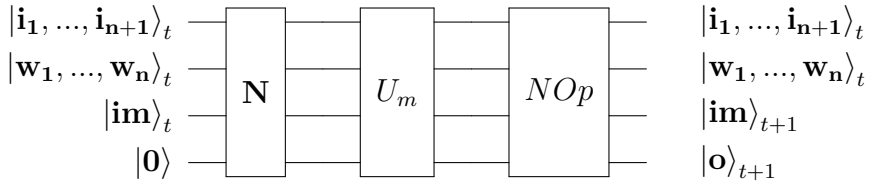
$$z(t+1) = N_z(z(t)\,|0\rangle + |1\rangle) \tag{1.14}$$

e

$$z(t+1) = N_z(z(t)\,|0\rangle + (1-z)\,|1\rangle) \tag{1.15}$$

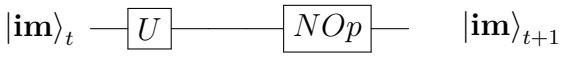
No experimento da dinâmica temporal do neurônio com apenas um qubit, testamos o operador de evolução unitária, o operador de BECHMANN-PASQUINUCCI; HUTTNER; GISIN, o operador de LEPORATI; ZANDRON; MAURI, uma adaptação do operador de ABRAMS; LLOYD, e o operador de medição quadrática (Quadratic Measurement nonlinear operator) descrito em NETO et al.. Experimentos verificando o valor do exponente de Lyapunov (STROGATZ, 1994) demonstraram que há presença de caos e bifurcação com os seguintes operadores acoplados: unitário, operador de BECHMANN-PASQUINUCCI; HUTTNER; GISIN e no operador descrito em (NETO et al., 2015). Detalhes desse experimento estão descritos no Apêndice C. Esses resultados indicam a possibilidade do neurônio de produzir comportamentos considerados

Figura 14 – Representação geral do neurônio quântico com memória interna.



Fonte: feito pelo autor.

Figura 15 – Neurônio quântico com memória interna de apenas um qubit acoplado com um operador não-linear.



Fonte: feito pelo autor.

atípicos dos sistemas dinâmicos, como, por exemplo, comportamentos caóticos e de bifurcação, e permitem que o neurônio explore comportamentos ditos naturais na biologia, como, por exemplo, armazenar informações processadas em seus instantes anteriores.

1.4.5 Comparação e contribuições

Nesta seção, vamos comparar o modelo de neurônio quântico proposto com os modelos existentes. Iremos nomear alguns modelos propostos intencionando facilitar a comparação. O nosso modelo proposto é chamado de qN, o modelo proposto por SCHULD; SINAYSKIY; PETRUCCIONE é chamado de qP, o modelo proposto por CAO; GUERRESCHI; ASPURU-GUZIK é chamado de qPNP, o modelo proposto por WIEBE; KLIUCHNIKOV é chamado de qPN, o modelo proposto por ALTAISKY é chamado de qAN, o modelo proposto por ZHOU et al. é chamado de qMPN e o modelo proposto por OLIVEIRA; SILVA; LUDERMIR é chamado de qRAM.

Podemos ver na Tabela 2 o comparativo dos modelos em relação às suas principais características. Vemos que o único neurônio que executa qualquer função não-linear é o modelo proposto. Os outros modelos simulam não linearidade, ou através de uma função não-linear específica, como a arcotangente (modelos qPNP e qPN), a função degrau (modelo qP) ou através de funções lógicas de uma única saída (modelo qRAM). A função que o neurônio computa está relacionada a sua complexidade computacional. Redes desses neurônios que executam funções de ativação degrau (modelo qP) só computam automâtos definidos (MINSKY, 1967). Redes Neurais que computam funções não-lineares como a sigmóide logística e a arcotangente (modelos qPNP e qPN) são capazes de simular Máquinas de Turing (SIEGELMANN; SONTAG, 1992). Uma camada de neurônios lógicos (qRAM) são capazes de simular Máquinas de Turing (OLIVEIRA; LUDERMIR, 1992). A vantagem de nosso modelo qN é que apenas um único neurônio é capaz de simular uma Máquina de Turing, considerando o uso da fita externa da Máquina à sua arquitetura, uma vez que ele possui vários qubits como saída. Os modelos qAN e qMPN são capazes de computar somatórios de funções unitárias e não-unitárias em seus processos de aprendizado, podendo transformar a dinâmica unitária do sistema em uma dinâmica quântica não-unitária. Essa não-unitariedade torna o modelo não-quântico (SILVA; OLIVEIRA; LUDERMIR, 2015).

O único modelo que cresce em espaço exponencialmente em função do tamanho da entrada é o modelo qRAM. Os modelos que não seguem uma complexidade temporal linear em função do tamanho da entrada (modelos qPNP e qPN) são os modelos que utilizam a arquitetura RUS (*repeat-until-success*), que podem precisar executar algumas vezes até que o resultado seja computado corretamente, o que torna também os modelos probabilísticos. Nosso modelo qN e os demais qP, qqAN e qMPN possui complexidade linear, uma vez que a quantidade de operações que ele irá executar é linear em função do tamanho da entrada.

O resultado da computação do neurônio pode estar na amplitude, no estado da base em um estado quântico ou em uma superposição de estados da base. Nos neurônios quânticos analisados, quatro modelos computam suas informações na amplitude do estado (modelos

qPNP, qPN, qAN, qMPN). O problema inerente de se processar informação na amplitude é que não é possível obter trivialmente essa informação a posteriori através de um operador de medição. Alguns neurônios (modelos qN, qP, qRAM) geram seu conteúdo em um estado da base de um estado quântico, o que torna o modelo com comportamento de circuito lógico. Os modelos qAN e qMPN podem ter seus resultados em superposição quântica. Nosso neurônio é o único que possui memória interna dinâmica, o que possibilidade a execução de funções não apenas espaciais como dependentes no tempo.

Utilizar modelos com memória interna explora a caoticidade presente nesses tipos de sistemas. O modelo de neurônio clássico proposto por AIHARA; TAKABE; TOYODA possui memória interna e apresenta caos e bifurcação devido a essa arquitetura de realimentação com informações temporais. Em nossos experimentos, podemos constatar a presença de caos, utilizando a métrica do expoente de Lyapunov, durante a dinâmica de neurônios simples com memória, *i.e.* sistemas quânticos com um único qubit operando sobre um operador unitário e um outro operador responsável por representar a interação com o ambiente, podendo ser unitário ou não-unitário. Mais detalhes desse experimento é encontrado no Apêndice - Parte C desta tese. É esperado que a descrição dos comportamentos caóticos, quando operadores não-unitários sejam utilizados, permita construir algoritmos de treinamento para redes neurais quânticas que explorem as regiões paramétricas onde o caos e bifurcação acontecem, uma vez que trabalhos anteriores têm encontrado correlação entre o aprendizado dos neurônios biológicos e a existência de bifurcação, mudança de fase e caos nas suas dinâmicas (HERZ et al., 2006; PASEMANN, 1993; GARLIAUSKAS, 1998; MATSUMOTO; TSUDA, 1988).

A principal vantagem do modelo quântico de neurônio proposto sobre os modelos clássicos existentes é a possibilidade de utilizar o paralelismo quântico e o emaranhamento. Isto significa que nós podemos alimentar a rede neural com todos os pesos possíveis e avaliá-los em superposição em uma única iteração (*i.e*, *single shot execution*). A quantidade de valores de pesos possíveis é muito grande, e esta tarefa de busca exaustiva seria impossível num computador clássico. É por isso que o processo de ajuste de pesos de uma rede neural segue heurísticas de minimização de erros ou algoritmos de otimização multiobjetiva YEUNG et al.**??**, e evolucionária TIAN et al..

Uma rede neural composta por neurônios quânticos pode executar o algoritmo backpropagation em uma versão quântica, uma vez que se esse algoritmo existe e é executado em um computador clássico (como função Booleana), ele também pode ser executado em um computador quântico. Existe sempre um operador quântico Booleano equivalente para um operador Booleano clássico (NIELSEN; CHUANG, 2000a). Porém, o desafio da computação quântica é ser capaz de recuperar informação pertinente a partir de uma operação quântica em superposição. Embora o operador quântico possa avaliar mais de uma entrada de uma única vez, não é possível checar todas as saídas em superposição. Apenas um valor clássico é verificado, probabilisticamente. Trabalhos anteriores tem mostrado o uso de operadores não-unitários para resolver problemas difíceis em computadores quânticos, cuja solução usando algoritmos

clássicos polinomiais não foi descoberta e ainda não se sabe da sua existência (ABRAMS; LLOYD, 1998; OHYA; VOLOVICH, 1999; NETO et al., 2015b; SILVA; OLIVEIRA; LUDERMIR, 2016a; PANELLA; MARTINELLI, 2011b). O aumento de velocidade na computação quântica obtido pela aplicação de operadores não-unitários é tido como um efeito não-físico embora em (CZACHOR, 1998a; CZACHOR, 1998b) seja mostrado uma versão de um operador deste tipo livre de influências não-físicas. Considerando a realizabilidade dos operadores quânticos não-unitários, podemos treinar uma rede neural quântica explorando todo espaço de parâmetros dos pesos, com as mais diversas topologias, em paralelo. Tal tarefa torna-se impossível de se realizar em computadores clássicos existentes.

Nós podemos usar o algoritmo de treinamento proposto por GAMMELMARK; MØLMER, que também foi utilizado para treinar redes sem-peso clássicas (SILVA; OLIVEIRA; LUDERMIR, 2016b). Podemos ajustar os valores dos pesos e a topologia da rede neural quântico em tempo polinomial.

Existem também algoritmos lineares quânticos definidos para treinar redes neurais. Esses algoritmos exploram o quântico paralelismo através do uso de algoritmos quânticos de busca (SILVA; OLIVEIRA; LUDERMIR, 2012; SILVA; OLIVEIRA; LUDERMIR, 2014). Trabalhos recentes tem mostrado que é possível selecionar arquiteturas de redes neurais usando memórias quânticas probabilísticas (SANTOS et al., 2018). Dessa forma, o resultado do treinamento da rede torna-se probabilístico em função da frequência dos bons resultados. Ainda assim, tais algoritmos são interessantes uma vez que explora todo o espaço de parâmetros e se existir valores de peso e arquiteturas ótimas, em alguma medição como resposta, tal arquitetura irá aparecer, desde que esse processo seja repetido um número limitado de vezes. Em algoritmos clássicos de treinamento, pode ser impossível transpor mínimos locais durante o período de ajuste de pesos. Estratégicas quânticas estendem a possibilidade de acessar valores paramétricos ótimos.

A realização do modelo proposto depende da existência de um computador quântico com a uma quantidade razoável de qubits disponíveis e de operadores quânticos. Os modelos quânticos computacionais disponíveis para simular tem apenas 20 qubits disponíveis. Avaliar por experimentos as ideias apresentadas neste trabalho em problemas reais requer um computador com habilidade de manipular centenas de qubits, o que é impossível ainda com a tecnologia corrente.

Tabela 2 – Tabela comparativa dos modelos de neurônios quânticos. O nosso modelo proposto é chamado de qN, o modelo proposto por SCHULD; SINAYSKIY; PETRUCCIONE é chamado de qP, o modelo proposto por CAO; GUERRESCHI; ASPURU-GUZIK é chamado de qPNP, o modelo proposto por WIEBE; KLIUCHNIKOV é chamado de qPN, o modelo proposto por ALTAISKY é chamado de qAN, o modelo proposto por ZHOU et al. é chamado de qMPN e o modelo proposto por OLIVEIRA; SILVA; LUDERMIR é chamado de qRAM.

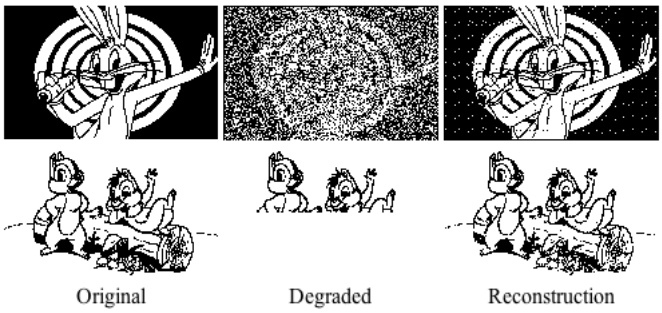
modelo de neurônio quântico	funções implement.	complex. espaço	complex. tempo	saída	memória interna dinâmica	prob / determ	unitário
qN	não-lineares	linear	linear	estado da base	x	d	x
qP	degrau	linear	linear	estado da base		d	x
qPNP	arcotangente periódica	linear	polin.	amplitude		p	x
qPN	arcotangente	linear	polin.	amplitude		p	x
qAN	unitária reversível e não-unitárias	linear	linear	estado superposto		d	
qMPN	unitária reversível e não-unitárias	linear	linear	estado superposto		d	
qRAM	não-lineares com uma saída	expon.	linear	estado da base		d	x

1.5 MEMÓRIAS ASSOCIATIVAS QUÂNTICAS

As redes neurais associativas funcionam recuperando informações parciais ou degradadas que foram previamente armazenadas em sua estrutura. As memórias associativas classicamente utilizam n neurônios para representar m padrões onde $m \geq 0.15n$ (HOPFIELD, 1982). O treinamento tradicional dessas redes envolvem a criação de bacias de atratores. Essas bacias de atratores permitem a convergência de informações semelhantes para uma mesma região, fazendo o processo exitoso de recuperação de informação degradada ou ruidosa (HOPFIELD, 1982). Um exemplo didático de entrada-saída de recuperação de informação é mostrado na Figura 16. A informação "original" é armazenada na rede através de um algoritmo de treinamento. Tradicionalmente, o algoritmo mais conhecido para fazer esse tipo de armazenamento é o algoritmo proposto por HOPFIELD. Após o armazenamento, uma dada entrada parcial ou com ruído pode ser recuperada na rede, em uma operação que se parece com uma memória tradicional, porém que permite encontrar informações com algum nível de diferença da informação armazenada.

As memórias quânticas quânticas utilizam outras estratégias para para armazenar e recuperar a informação. Algumas das soluções utilizam a *superposição quântica* para armazenar 2^n padrões usando n neurônios (VENTURA; MARTINEZ, 1999; ZHOU et al., 2012b), o que permite um ganho exponencial de armazenamento em relação às alternativas clássicas (HOPFIELD, 1984). Para o

Figura 16 – Exemplo de funcionamento das redes de memória associativa. Na primeira coluna à esquerda, mostra-se duas figuras que se deseja armazenar na rede. Na segunda coluna, estão as figuras degradadas, com ruído ou parciais. Na terceira coluna, as figuras recuperadas/reconstruídas pela rede a partir das figuras degradas.



Fonte: http://www.decom.ufop.br/imobilis/redes-de-hopfield/

processo de recuperação da informação do conteúdo armazenado, há diversas estratégias. O algoritmo de busca de *Grover* tem sido usado para amplificar as amplitudes dos estados da base esperados em um dado estado quântico. VENTURA; MARTINEZ propuseram uma modificação do algoritmo de busca de Grover em que o operador quântico de busca (oráculo) depende do estado em que se deseja verificar a existência dele na memória, aqui chamado de **estado-pergunta**, e dos estados armazenados. Isso significa que a arquitetura do circuito é dependente e deve ser reconfigurada para cada estado-pergunta e para cada conjunto de treinamento. EZHOV; NIFANOVA; VENTURA estende o trabalho de Ventura e Martinez para incluir algum nível de tolerância da distância entre o estado-pergunta e o conteúdo salvo na memória. TRUGENBERGER propõe uma memória probabilística que calcula a distância de Hamming do estado pergunta e os estados salvos na memória utilizando um operador Hamitoniano. Em (ZHOU et al., 2012b), um modelo de memória utiliza um operador não-linear para fazer a recuperação da informação.

1.6 CONTRIBUIÇÕES EM REDES NEURAIS ASSOCIATIVAS QUÂNTICAS

Propomos aqui dois modelos de memórias quânticas associativas. A primeira delas utiliza um operador não-unitário para fazer a recuperação da informação armazenada. Essa memória, através do uso do operador não-unitário, possui custo constante de acesso à informação, mas, como está descrita, possui a limitação de não se ter uma prova da sua realizabilidade física [7]. A segunda memória proposta utiliza operadores unitários para fazer tal recuperação, envolvendo a TQF e o AG. As vantagens e limitações dessas memórias serão apresentadas nessa seção.

[7] Embora não se tenha uma prova de sua realizabilidade física, tal operador pode ser comparado com outros operadores quânticos existentes do ponto de vista de complexidade de circuito. Em outras palavras, é possível verificar o limite superior da quantidade de passos necessários no algoritmo que utiliza tal operador.

1.6.1 Memória quântica associativa não-linear

Em 2012, ZHOU et al. propõem a utilização de um operador não-unitário para recuperação de informação em uma memória quântica. Esse operador reduz a busca para um custo de tempo linear. Ele foi utilizado em trabalhos anteriores, resolvendo problemas NP-Completos em uma arquitetura quântica em tempo polinomial (NETO et al., 2015b; LEPORATI; FELLONI, 2007). Neste trabalho, utilizamos um operador não-unitário proposto em (LEPORATI; FELLONI, 2007) para recuperar a informação em uma memória quântica associativa, e reduzimos o custo dessa operação para uma constante. Neste algoritmo, o operador é capaz de fazer uma busca e modificar o estado de um registrador caso encontre um determinado comportamento em um estado quântico. Este operador não-unitário O_f é descrito abaixo.

$$O_f = 2^n \begin{bmatrix} 0 & 0 \\ & \\ 0 & 1 \end{bmatrix} = 2^n \left|1\right\rangle \left\langle 1\right| . \tag{1.16}$$

Neste caso, para entradas superpostas onde há valores não-nulos para a amplitude do estado da base $|1\rangle$ o resultado será o estado da base $|1\rangle$ com a amplitude amplificada. Caso contrário, o que será verificado na saída será um vetor nulo.

$$O_f \left|0\right\rangle = \mathbf{0}$$

e

$$O_f \left|1\right\rangle = 2^n \left|1\right\rangle$$

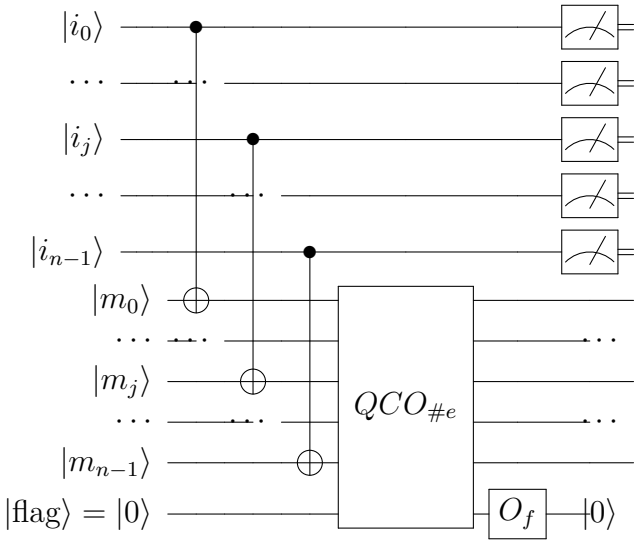
$$O_f(\alpha \left|0\right\rangle + \beta \left|1\right\rangle) = 2^n \beta \left|1\right\rangle .$$

Um operador quântico unitário $QCO_{\#e}$ também é utilizado na descrição do circuito proposto. Esse operador dá como saída 1 no registrador $|\text{flag}\rangle$ se existe pelo menos e qubits iguais entre um padrão na memória e conteúdo de entrada.

O conteúdo da memória é armazenado em um estado quântico superposto $|m_0, ..., m_{n-1}\rangle$. Para uma dada entrada, a memória realiza um cálculo de distância usando uma sequência de operadores Não-Controlados, como mostrado na Figura 17. Caso um qubit da entrada seja igual a um qubit da memória, tal qubit da memória correspondente será zerado. O operador $QCO_{\#e}$ então verifica se a quantidade de elementos zerados é igual ou maior que um valor e pré-estabelecido. Caso for verdade, ele modifica o valor do registrador $|flag\rangle$ para 1. O operador O_f então faz a identificação dos estados em que o valor da flag é $|1\rangle$. A recuperação do conteúdo então se dá no primeiro registrador como uma leitura simples de medição.

Por não possuir uma implementação física existente do operador não unitário O_f, este modelo de memória ainda é um modelo teórico de recuperação de conteúdo. Há alguma evidência de que algoritmos não-unitários quânticos são equivalentes a algoritmos quânticos probabilísticos (GINGRICH; WILLIAMS, 2004). Trabalhos futuros podem estar direcionados no

Figura 17 – Circuito da memória quântica associativa não-linear. A entrada possui n qubits, $|i_0\rangle ... |i_{n-1}\rangle$, e a memória é armazenada nos n qubits $|m_0\rangle ... |m_{n-1}\rangle$. O resultado é verificado no último qubit $|flag\rangle$. Para uma dada entrada, caso exista um padrão igual ou com e qubits iguais ao conteúdo armazenado na memória, o último qubit poderá ser lido com o valor 1. Caso contrário, o valor a ser lido é vazio.



Fonte: feito pelo autor.

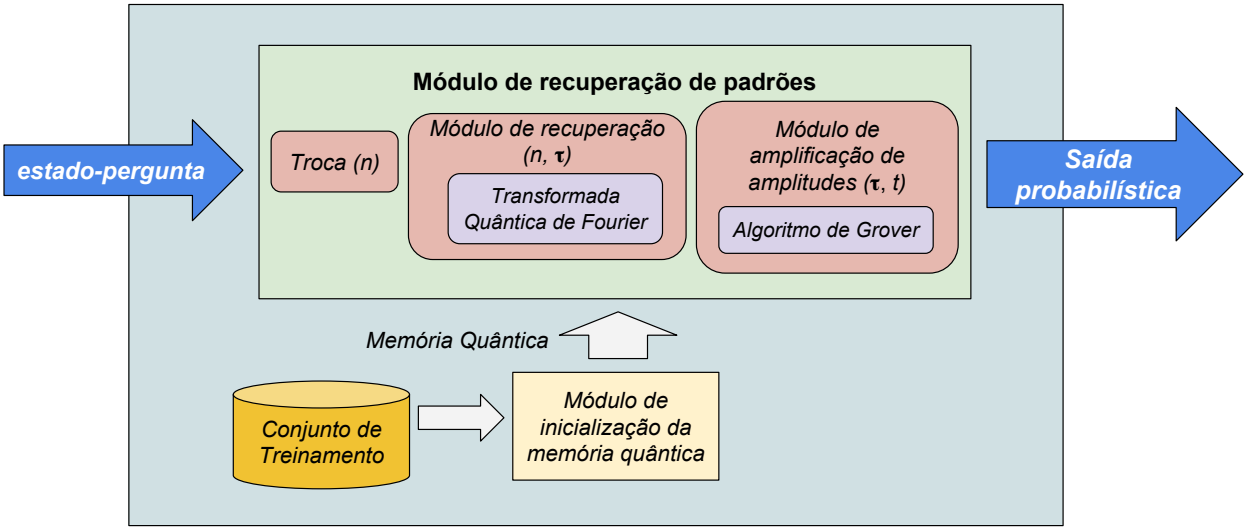
estudo aprofundado de um algoritmo quântico probabilístico equivalente na utilização desta memória proposta. Mais detalhes da implementação deste algoritmo em um circuito quântico é dado no Apêndice D.

1.6.2 Memória quântica associativa linear

A memória quântica associativa linear proposta também armazena a informação em um registrador quântico em superposição. Uma informação de entrada (estado-pergunta) tem seu conteúdo comparado com o conteúdo da memória usando o algoritmo de estimativa de fase associada à TQF. Após esse cálculo, as amplitudes dos estados de menor distância, com algum nível de tolerância, são amplificadas usando o algoritmo de busca AG. A arquitetura da memória está descrita na Figura 18.

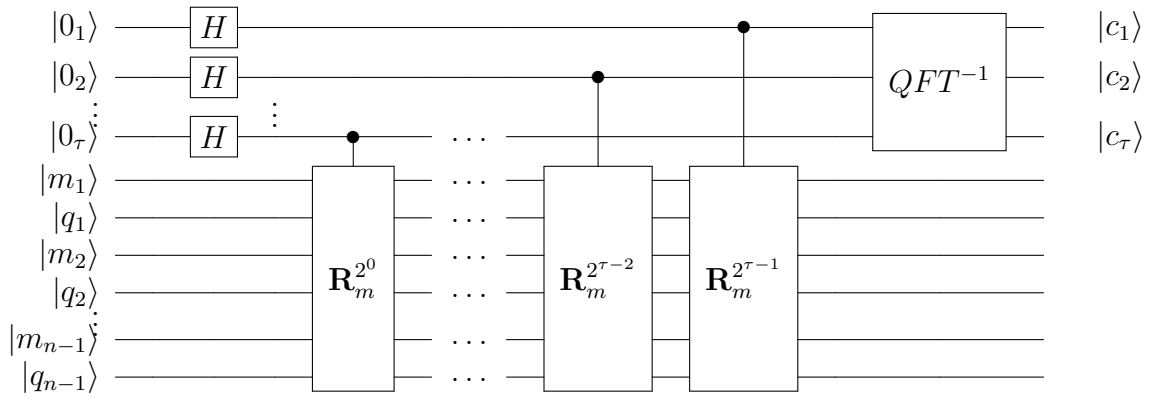
O circuito que calcula a distância de Hamming de uma dado estado-pergunta com o conteúdo da memória está descrito na Figura 19. Há uma sequência de τ operadores $\mathbf{R}_m$-Controlados que calculam essa distância, colocando a informação na fase, de maneira que a transformada quântica inversa de Fourier, QFT^{-1}, consiga recuperar essa informação que está na fase. O operador $\mathbf{R}_m$ está descrito na Equação 1.17. O valor de τ é a quantidade de qubits do registrador quântico que irá armazenar o resultado da distância. Mais detalhes da implementação deste algoritmo em um circuito quântico é dado no Apêndice E.

Figura 18 – Arquitetura da Memória Quântica Associativa.



Fonte: feita pelo autor.

Figura 19 – Detalhamento do circuito da memória quântica associativa linear proposta. Este circuito faz a comparação entre um estado quântico de entrada $|q_1, ..., q_n\rangle$ com cada um do conteúdo na memória $|m_1, ..., m_n\rangle$. O circuito adiciona na fase um fator multiplicativo em função de quão próximo está a entrada do conteúdo na memória através do operador $\mathbf{R}_m$. A sequência de $\mathbf{R}_m$-Controlados permite deixar o conteúdo da fase no formato em que a transformada inversa de Fourier QFT^{-1} consiga recuperar essa informação e transformá-la em um estado quântico da base correspondente à distância calculada.



Fonte: feito pelo autor

$$R_m = \begin{pmatrix} e^{2\pi i \Delta d} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i \Delta d} \end{pmatrix} \tag{1.17}$$

Para uma dada distância aceitável do conteúdo da entrada com os valores armazenados na memória, é possível realizar a amplificação das amplitudes dos estados da base que representam essa distância tolerável, usando um algoritmo de Grover modificado (BIRON et al., 1999). Tal algoritmo de Grover modificado permite amplificar as amplitudes mesmo que o estado quântico envolvido não esteja completamente superposto.

1.6.3 Comparação e contribuições

Nesta seção, faremos a comparação dos modelos propostos com os modelos existentes. Iremos nomear os modelos com o objetivo de facilitar a comparação. O modelo de memória quântica não-linear proposto é chamado de qMN, o modelo de memória quântica linear proposto é chamado de qM. O modelo de memória proposto por VENTURA; MARTINEZ é chamado de qMV, o modelo proposto por TRUGENBERGER é chamado de qMT, o modelo proposto por EZHOV; NIFANOVA; VENTURA é chamado de qME e o modelo proposto por NJAFA; ENGO; WOAFO é chamado de qMNN. A Tabela 3 compara os modelos de acordo com suas principais características.

As memórias quânticas têm complexidade de execução linear (modelos qM, qMV, qME, qMT e qMNN), ou seja, são modelos iterativos em função do tamanho da entrada. O modelo proposto qMN tem complexidade temporal constante, porém sua execução depende da implementação de seu operador não-unitário.

Dos modelos analisados, apenas o modelo qMVM não possui capacidade de generalizar o processo de recuperação para uma entrada próxima ao conteúdo armazenado ou com ruído. Os modelos propostos qMN e qM e o modelo qMT são os únicos que possuem arquitetura fixa em função do tamanho da entrada e do conteúdo a ser armazenado. Os outros modelos (qMVM, qME e qMNN) possuem arquitetura variável em função do conjunto de treinamento. Ou seja, operadores unitários desses modelos são construídos a depender do conteúdo da memória e/ou do conteúdo de entrada ou *estado-pergunta*.

Os modelos qM, qMVM, qME e qMNN são baseados no algoritmo de busca de Grover e por isso recuperam informação probabilisticamente com a possibilidade de dar como saída um conteúdo que não estava armazenado na memória. Esses estados não armazenados são chamados de estados-espúrios. Eles aparecem devido ao processo de amplificação dos estados que antes possuíam valor de amplitude igual a zero. O único modelo que possui robustez a estados espúrios é o modelo qMT. Dentre os modelos existentes, apenas os modelos propostos qM e qMN possui a capacidade de escolha de quão distante o estado-pergunta pode estar do

conteúdo armazenado, incluindo um grau de tolerância no seu processo de recuperação.

A contribuição principal do nosso trabalho está na combinação das características principais das memórias existentes. Nossos modelos são capazes de executar linearmente ou em tempo constante a recuperação de informação na memória, permitir entradas com algum nível de diferença em relação ao conteúdo armazenado, com a escolha desse nível de tolerância, além de possuir uma arquitetura estática, independente do conteúdo armazenado. A vantagem de possuir uma arquitetura estática está na dificuldade em implementar circuitos quânticos gerais (KNILL, 2008). Nós não conseguimos reduzir ou eliminar a presença dos estados espúrios, única vantagem que não está presente em nossos modelos, mas que também está presente na memória qMT.

Modelos de memória quântica também podem ser utilizados na escolha de arquitetura de redes neurais clássicas ou quânticas (SANTOS et al., 2018). Através dos modelos de memória quântica associativa propostos, podemos utilizá-los como base na construção de novos algoritmos de escolha de arquitetura, sendo um diferencial em resultado se a mudança de arquitetura da rede neural tiver impacto considerável na eficiência da rede. O modelo de memória utilizado no algoritmo de definição de arquitetura em SANTOS et al. utiliza a memória qMV que possui dificuldade em recuperação de informação quando o conteúdo a ser recuperado é muito diferente dos padrões armazenados na memória. Se a rede neural tiver eficiência muito sensível à arquitetura, *i.e* para diferentes topologias de arquiteturas a eficiência da rede varia com significância definida, a aplicação da memória qMV para definir essa arquitetura vai ter pouca eficiência porque o resultado estará fragmentado nos diversos estados a serem recuperados. Nosso modelo qM pode ser indicado a ter melhor performance uma vez que utilizamos o algoritmo de Grover adaptado, que funciona cada vez melhor quando os estados presentes possuem poucos valores nulos. Por exemplo, dado o conjunto de treinamento com padrões muito distintos, a memória do (TRUGENBERGER, 2002b) não funciona bem. Por exemplo, dado um conjunto de treinamento $|0000001\rangle, |1111111\rangle, |1111110\rangle, |1111001\rangle$, e um estado-pergunta $|0000001\rangle$, a memória do TRUGENBERGER recupera com 27% de probabilidade. Nossa memória proposta o faz com 56.6%. Isso se dá porque após o processamento da transformada quântica de Fourier, o estado $|111\rangle|0000001\rangle|0000001\rangle + |001\rangle|0000001\rangle|1111111\rangle + |000\rangle|0000001\rangle|1111110\rangle + |011\rangle|0000001\rangle|1111001\rangle$ será processado pelo algoritmo de Grover, que irá fazer a recuperação dos três primeiros qubits iguais a $|111\rangle$, se $t = 0$. Com 10 iterações, a amplitude desse estado é ampliado para 0.75244140625, o que dá 56% de chance de recuperar esse estado.

Trabalhos futuros podem estar voltados na avaliação prática da aplicação de nossas memórias em bases de dados reais e no estudo da realizabilidade do operador quântico não-unitário em termos de computação quântica probabilística.

Tabela 3 – Tabela comparativa dos modelos de memórias associativas quânticas. O nosso modelo proposto de memória quântica não-linear é chamado de qMN, e o modelo de memória quântica linear proposto é chamado de qM. O modelo de memória proposto por VENTURA; MARTINEZ é chamado de qMV, o modelo proposto por TRUGENBERGER é chamado de qMT, o modelo proposto por EZHOV; NIFANOVA; VENTURA é chamado de qME e o modelo proposto por NJAFA; ENGO; WOAFO é chamado de qMNN.

modelos de memória	complexidade temporal	permite entradas com ruído	arquitetura estática	não é sensível a estados espúrios	possui variável de grau de tolerância
qMN	constante*	x	x		x
qM	linear	x	x		x
qMV	linear				
qME	linear	x			
qMT	linear	x	x	x	
qMNN	linear	x			

1.7 PUBLICAÇÕES

Os trabalhos completos publicados[8] durante a realização do doutorado de Fernando M de Paula Neto iniciado em Março de 2016 estão listados abaixo:

- **DE PAULA NETO, F. M.**; DE OLIVEIRA, WILSON R. ; LUDERMIR, TERESA B; DA SILVA, ADENILTON J. *Chaos in a quantum neuron: An open system approach*. NEUROCOMPUTING, v. 246, p. 3-11, 2017.

- CAMBUIM, LUCAS F.S. ; MACIEIRA, RAFAEL M. ; **DE PAULA NETO, F. M.** ; BARROS, EDNA ; LUDERMIR, TERESA B. ; ZANCHETTIN, CLEBER . *An Efficient Static Gesture Recognizer Embedded System Based on ELM Pattern Recognition Algorithm*. Journal of Systems Architecture, v. 68, p. 1, 2016.

- PINHEIRO, HECTOR N. B. ; **DE PAULA NETO, F. M.**; OLIVEIRA, ADRIANO L. I. ; REN, TSANG ING ; CAVALCANTI, GEORGE D. C. ; ADAMI, ANDRE G. . *Optimizing speaker-specific filter banks for speaker verification*. In: ICASSP 2017 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, New Orleans. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017. p. 5350.

- **DE PAULA NETO, F. M.** OLIVEIRA, Wilson Rosa ; DA SILVA, ADENILTON JOSE ; LUDERMIR, TERESA BERNARDA . *On the Entanglement Dynamics of the Quantum Weightless Neuron*. In: 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017, Uberlândia. 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017. p. 175.

[8] Aqui estão listados também os trabalhos que não possuem conexão direta com esta tese.

- **DE PAULA NETO, F. M.**; DA SILVA, ADENILTON JOSE ; DE OLIVEIRA, WILSON ROSA ; LUDERMIR, TERESA BERNARDA . *Non-unitary Quantum Associative Memory*. In: 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017, Uberlândia. 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017. p. 97.
- **DE PAULA NETO, F. M.**; LUDERMIR, TERESA B; DE OLIVEIRA, WILSON R. ; DA SILVA, ADENILTON J. *Quantum Perceptron with Dynamic Internal Memory*. In: International Joint Conference on Neural Network, 2018, Rio de Janeiro. Proceedings of International Joint Conference on Neural Network. Los Alamitos: IEEE, 2018. v. 1. p. 1-8.

1.8 DETALHAMENTO DA TESE

O detalhamento dos modelos propostos desta Tese se encontra no Apêndice. A forma de apresentação é através de artigos publicados ou submetidos a revistas/conferências da área.

- Apêndice Parte A - **DE PAULA NETO, F. M.**; Teresa B. Ludermir ; De Oliveira, W.R. ; DA SILVA, ADENILTON J. *Quantum Perceptron with Dynamic Internal Memory*. In: International Joint Conference on Neural Network, 2018, Rio de Janeiro. Proceedings of International Joint Conference on Neural Network. Los Alamitos: IEEE, 2018. v. 1. p. 1-8.
- Apêndice Parte B - **DE PAULA NETO, F. M.**; Teresa B. Ludermir ; De Oliveira, W.R. ; DA SILVA, ADENILTON J. *Implementing nonlinear activation functions for quantum neurons* (submetido para a revista IEEE Transactions on Neural Networks and Learning Systems)
- Apêndice Parte C - **DE PAULA NETO, F. M.**; Teresa B. Ludermir ; De Oliveira, W.R. *Quantum neurons with internal memory* (preparado para ser submetido para a revista Neurocomputing).
- Apêndice Parte D - **DE PAULA NETO, F. M.**; DA SILVA, ADENILTON JOSE ; DE OLIVEIRA, Wilson Rosa ; LUDERMIR, TERESA BERNARDA . *Non-unitary Quantum Associative Memory*. In: 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017, Uberlândia. 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017. p. 97.
- Apêndice Parte E- **DE PAULA NETO, F. M.**; DA SILVA, ADENILTON J. ; De Oliveira, W.R. ; LUDERMIR, Teresa B . *Quantum probabilistic associative memory architecture* (submetido para a revista Neurocomputing).

Na Parte A, o perceptron quântico com memória interna é detalhado. Na Parte B, o modelo geral de neurônio quântico que executa qualquer função de ativação é detalhado. Em seguida,

na Parte C, é descrito o modelo de neurônio que executa qualquer função de ativação não-linear, que possui memória interna e que pode estar acoplado com operadores não-unitários. Neste trabalho, experimentos mostram que operadores unitários acoplados com operadores unitários emulam comportamento caótico e de bifurcação.

Na Parte D, um algoritmo de recuperação não-linear para memória quântica é descrito. Em seguida, na Parte E, um algoritmo unitário de recuperação para memória quântica é detalhado. Esse modelo utiliza os algoritmos quânticos de Grover e da transformada inversa de Fourier no seu processo de recuperação.

1.9 CONCLUSÃO

Neurônios artificiais são modelos importantes na área de aprendizagem de máquina. A literatura de neurônios quânticos não contava com a presença de modelos que incluíssem memória interna ou que pudessem realizar qualquer função de ativação não-linear. Ou seja, não existia um modelo de neurônio quântico que pudesse simular o comportamento de um neurônio clássico genérico e que mantivesse as propriedades quânticas durante sua execução com um custo de execução linear. Este trabalho contribuiu propondo modelos de neurônios que são capazes de simular os neurônios clássicos (com a descrição da implementação de funções de ativação quaisquer em um circuito quântico com custo linear), utilizar os recursos quânticos de paralelismo e emaranhamento, possuir acoplamento com operadores não-unitários e memória interna dinâmica. Com a proposição do modelo descrito neste trabalho, trabalhos futuros podem descrever algoritmos quânticos de treinamento para redes neurais quânticas dinâmicas ou estáticas e utilizar do paralelismo e emaranhamento (JOZSA; LINDEN, 2003; JOZSA, 1997), que tem sido apontado como o diferencial dos sistemas quânticos.

Em termos das memórias associativas, pudemos propor dois métodos diferentes de recuperação de informação em uma memória associativa. Um desses métodos envolve a captura da informação através de um operador não-linear quântico. Este operador permite a recuperação de informação em um número constante de passos, porém sua implementação está dependente da realizabilidade do operador não-unitário utilizado. O outro método recuperação de informação quântica é unitário e envolve o uso do cálculo de uma distância euclidiana parametrizável utilizando a transformada inversa de Fourier e do algoritmo de Grover para fazer a amplificação dos estados da base que representam baixa distância para uma dada entrada-pergunta. As memórias propostas possuem arquitetura fixa independente do estado-pergunta e do conteúdo armazenado na memória, possuindo custo polinomial de recuperação de informação.

1.9.1 Principais resultados e trabalhos futuros

Os principais resultados desenvolvidos durante a execução dessa tese e possíveis trabalhos futuros são detalhados abaixo para cada um dos artigos que a compõem.

- Propusemos a descrição de um Perceptron quântico com memória interna. Isso significa

que o neurônio é capaz de simular funções não só espaciais mas temporais. No trabalho, é possível verificar três principais tipos de Perceptron com memória: (1) um perceptron quântico cuja saída é o novo estado da memória interna, (2) um perceptron quântico cuja memória interna pode ser alterada por operadores quânticos reversíveis e (3) um perceptron quântico cuja memória interna pode ser alterada por qualquer operador quântico unitário. A proposição desses três modelos tem a ver com o grau de memória disponível e a capacidade de alterar internamente sua informação. A proposição desse modelo permite incluir na dinâmica neuronal quântica comportamentos encontrados em alguns neurônios biológicos e em modelos matemáticos que se aproxima do comportamento do neurônio humano (PASEMANN, 1997), como por exemplo caos. Estes resultados estão detalhados no artigo presente no Apêndice - Parte A. Trabalhos futuros podem ser direcionados na proposição de novos modelos quânticos de aprendizagem que permitem a utilização das vantagens da computação quântica (BIAMONTE et al., 2017b), como o emaranhamento, como aspecto indutor ao aprendizado (JOZSA; LINDEN, 2003; JOZSA, 1997; NETO et al., 2017; ŻYCZKOWSKI et al., 2001). Há modelos de neurônios quânticos que não possuem ainda algoritmo de treinamento (SCHULD; SINAYSKIY; PETRUCCIONE, 2014b; NETO et al., 2018; CAO; GUERRESCHI; ASPURU-GUZIK, 2017), assim como algoritmos de treinamento para redes neurais quânticas que que não considera a adaptação dos pesos das unidades de processamento que estão nas camadas intermediárias (SILVA; OLIVEIRA; LUDERMIR, 2012). Resolver limitações de modelos existentes é fundamental para o avanço da área de pesquisa e para as sucessivas contribuições;

- Um neurônio quântico é descrito para representar qualquer função de ativação não-linear utilizando o algoritmo de estimativa de fase e a transformada quântica de Fourier. Com esta proposição, uma arquitetura quântica de redes neurais artificiais de várias camadas podem ser implementadas em um circuito quântico. Utilizando os algorimos quânticos de treinamento (unitários e não-unitários) existentes, é possível explorar todo o espaço de busca dos parâmetros envolvidos. Exemplos são dados de funções de ativação lineares, sigmoide e de base radial. Estes resultados estão detalhados no artigo presente no Apêndice - Parte B. Nessa direção, os trabalhos futuros estão relacionados à proposição de algoritmos de treinamento lineares e não-lineares que levem em conta a arquitetura, funções de ativação, pesos e outros parâmetros da rede neural quântica. O impacto da utilização do emaranhamento no processo de treinamento também é um assunto em aberto. Também parece promissor processar informações na amplitude dos estados quânticos ao invés de utilizar os estados da base para representar informação. Há trabalhos que calculam funções de distância de Hamming dentro desse paradigma chamado *amplitude encoding* (SCHULD; FINGERHUTH; PETRUCCIONE, 2017). É também promissor investigar como os operadores que modificam a fase de um estado quântico podem ser reconstruídos minimamente usando operadores de mudança de fase simples, como os operadores quânticos da família de Pauli, uma vez que os modelos propostos de

neurônio quântico utilizam operadores que mudam o valor da fase.

- Propusemos um modelo de neurônio que possui qualquer função de ativação em seu funcionamento e que possa também ter memória interna. Nesse modelo também é possível acoplar o neurônio com algum operador de interação com o ambiente. Os operadores de interação com o ambiente utilizados foram operadores unitários e não-unitários existentes na literatura e utilizados para resolver outros tipos de problema. Esse modelo é então a generalização do modelo que propusemos do perceptron com memória em *Quantum Perceptron with Dinamic Internal Memory*, com o modelo de neurônio que calcula qualquer função de ativação apresentado em *Implementing nonlinear activation functions for quantum neurons*. No estudo, foi feito um experimento dos operadores lineares e não-lineares acoplados com operadores unitários numa perspectiva iterativa inspirada no trabalho do KISS et al.. O diagrama de órbitas dessa dinâmica apresentou resultados interessantes em termos de comportamentos caóticos e de bifurcação. Os valores do expoente de Lyapunov indicaram a presença de caos na dinâmica de operadores unitários e não-unitários. Estes resultados estão detalhados no artigo presente no Apêndice - Parte C. Trabalhos futuros envolvem o detalhamento dessa dinâmica e o estudo de como a utilização desses operadores podem modificar a aprendabilidade dos neurônios. A proposição de algoritmos de treinamento quânticos que sejam compatíveis com esse modelo e que utilizem recursos intrínsecos da computação como emaranhamento também são promissores (JOZSA, 1997). Também é possível fazer uma esquematização geral dos operadores não-unitários, sob a análise de computabilidade, complexidade temporal e espacial. É possível a partir de um esquema como esse, construir algoritmos e circuitos quânticos que possam usar ou não operadores não-unitários a partir de sua necessidade, limitações de hardware ou complexidade; É também promissor fazer a identificação de padrões de comportamento entre aprendizagem e os parâmetros de modelos de redes. Nós iniciamos esse estudo paramétrico da dinâmica da rede neural quântica (NETO et al., 2015; NETO et al., 2015a; PAULA, 2016; NETO et al., 2017). Espera-se utilizar esse conhecimento na descrição de modelos mais robustos de aprendizagem clássica e quântica, considerando e verificando o impacto das dinâmicas caóticas na descrição desses modelos.

- Propousemos a descrição de um modelo de memória utilizando um operador não-unitário, que permite a captura de informação quântica de forma não-iterativa. Se for possível implementar tal operador e checar sua saída, é possível checar a presença de uma entrada-pergunta na memória em um número constante de passos. A limitação do modelo está na dependência de que esse operador não-unitário possa ser implementado em dispositivos físicos. Há alguns experimentos que demonstram a possibilidade de implementar não-unitariedade quântica usando algoritmos quânticos probabilísticos (GINGRICH; WILLIAMS, 2004). Estes resultados estão detalhados no artigo presente no Apêndice - Parte D. Trabalhos futuros podem ser guiados para o detalhamento desses modelos incluindo sua

equivalência, ou proximidade, a modelos cuja implementação física é conhecida.

- No trabalho *Quantum probabilistic associative memory architecture*, nós propomos a utilização da transformada inversa de Fourier quântica para calcular a distância de um estado-pergunta para os estados superpostos salvos na memória. O cálculo dessa distância é feito qubit a qubit entre cada estado salvo na memória e o estado-pergunta. A vantagem deste modelo proposto é que possui arquitetura estática, que independente do conjunto de treinamento e do estado-pergunta. O algoritmo de Grover modificado é usado para amplificar as amplitudes dos estados de interesse, que possuem menos qubits que os outros modelos existentes. No modelo proposto, é possível fazer a recuperação de estados quânticos com taxa de acerto maior se o processo de recuperação puder ser repetido algumas vezes. A limitação dessa proposta é que o algoritmo de Grover inclui estados espúrios, que não estavam na base de dados, como possíveis estados a serem capturados na saída. Estes resultados estão detalhados no artigo presente no Apêndice - Parte E. Trabalhos futuros podem incluir a resolução de problema dos modelos existentes, como os modelos baseados no Algoritmo de Grover (VENTURA; MARTINEZ, 1999), que insere estados indesejados e espúrios na recuperação de informação quântica, ou ainda os que possuem comportamento indesejado para uma base de treinamento não-esparsa (TRUGENBERGER, 2002b). É um trabalho futuro muito promissor o estudo ou proposições de versões do algoritmo de Grover que minimizem a presença dos estados espúrios e que aumentem a probabilidade de verificar os estados esperados, uma vez que há várias memórias quânticas que utilizam esse algoritmo.

REFERÊNCIAS

ABRAMS, D. S.; LLOYD, S. Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems. *Phys. Rev. Lett.*, v. 81, n. 18, p. 3992–3995, 1998.

AIHARA, K.; TAKABE, T.; TOYODA, M. Chaotic neural networks. *Physics Letters A*, v. 144, n. 6, p. 333 – 340, 1990. ISSN 0375-9601.

ALMEIDA, L. M.; LUDERMIR, T. An improved method for automatically searching near-optimal artificial neural networks. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. [S.l.: s.n.], 2008. p. 2235–2242. ISSN 2161-4393.

ALMEIDA, L. M.; LUDERMIR, T. B. A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks. *Neurocomputing*, v. 73, n. 7, p. 1438 – 1450, 2010. ISSN 0925-2312. Advances in Computational Intelligence and Learning. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231209004135>.

ALTAISKY, M. V. *Quantum neural network*. Russia, 2001.

ASADUZZAMAN, M.; UDDIN, A. N.; SHAHJAHAN, M.; MURASE, K. Harnessing chaotic activation functions in training neural network. In: SPRINGER. *International Conference on Neural Information Processing*. [S.l.], 2012. p. 551–558.

BECHMANN-PASQUINUCCI, H.; HUTTNER, B.; GISIN, N. Non-linear quantum state transformation of spin-1/2. *Physics Letters A*, v. 242, p. 198–204, 1998.

BEHRMAN, E.; NASH, L.; STECK, J.; CHANDRASHEKAR, V.; SKINNER, S. Simulations of quantum neural networks. *Information Sciences*, v. 128, n. 3–4, p. 257 – 269, 2000.

BIAMONTE, J.; WITTEK, P.; PANCOTTI, N.; REBENTROST, P.; WIEBE, N.; LLOYD, S. Quantum machine learning. *Nature*, Macmillan Publishers Limited, part of Springer Nature. All rights reserved. SN -, v. 549, p. 195 EP –, Sep 2017.

BIAMONTE, J.; WITTEK, P.; PANCOTTI, N.; REBENTROST, P.; WIEBE, N.; LLOYD, S. Quantum machine learning. *Nature*, Nature Publishing Group, v. 549, n. 7671, p. 195, 2017.

BIRON, D.; BIHAM, O.; BIHAM, E.; GRASSL, M.; LIDAR, D. A. Generalized grover search algorithm for arbitrary initial amplitude distribution. In: WILLIAMS, C. P. (Ed.). *Quantum Computing and Quantum Communications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. p. 140–147. ISBN 978-3-540-49208-5.

BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: Livros Técnicos e Científicos, 2000.

BREUER, H.-P.; PETRUCCIONE, F. *The theory of open quantum systems*. [S.l.]: Oxford University Press on Demand, 2002.

CAO, Y.; GUERRESCHI, G. G.; ASPURU-GUZIK, A. Quantum neuron: an elementary building block for machine learning on quantum computers. *arXiv preprint arXiv:1711.11240*, 2017.

CZACHOR, M. Notes on nonlinear quantum algorithms. *arXiv preprint quant-ph/9802051*, 1998.

CZACHOR, M. Remarks on search algorithms and nonlinearity. *acta physica slovaca*, SLOVAK ACADEMY OF SCIENCES, v. 48, p. 157–162, 1998.

DEUTSCH, D. Quantum theory, the church-turing principle and the universal quantum computer. In: THE ROYAL SOCIETY. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. [S.l.], 1985. v. 400, n. 1818, p. 97–117.

DIRAC, P. A. M. A new notation for quantum mechanics. In: CAMBRIDGE UNIVERSITY PRESS. *Mathematical Proceedings of the Cambridge Philosophical Society*. [S.l.], 1939. v. 35, n. 3, p. 416–418.

EZHOV, A. A.; NIFANOVA, A. V.; VENTURA, D. Quantum associative memory with distributed queries. *Inf. Sci. Inf. Comput. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 128, n. 3-4, p. 271–293, out. 2000. ISSN 0020-0255.

FERNANDEZ, B.; PARLOS, A. G.; TSAI, W. K. Nonlinear dynamic system identification using artificial neural networks (anns). In: *1990 IJCNN International Joint Conference on Neural Networks*. [S.l.: s.n.], 1990. p. 133–141 vol.2.

FEYNMAN, R. P. Simulating physics with computers. *International journal of theoretical physics*, Springer, v. 21, n. 6-7, p. 467–488, 1982.

FICHTNER, K.-H.; INOUE, K.; OHYA, M. On a quantum model of brain activities — distribution of the outcomes of eeg-measurements and random point fields. *Open Systems & Information Dynamics*, v. 19, n. 04, p. 1250025, 2012.

FREEMAN, W. J. Tutorial on neurobiology: From single neurons to brain chaos. *International Journal of Bifurcation and Chaos*, v. 02, n. 03, p. 451–482, 1992. Disponível em: <https://doi.org/10.1142/S0218127492000653>.

GAMMELMARK, S.; MØLMER, K. Quantum learning by measurement and feedback. *New Journal of Physics*, IOP Publishing, v. 11, n. 3, p. 033017, 2009.

GARLIAUSKAS, A. Neural network chaos analysis. *Nonlinear Anal., Model. and Control*, v. 3, p. 43–57, 1998.

GINGRICH, R. M.; WILLIAMS, C. P. Non-unitary probabilistic quantum computing. In: *Proceedings of the Winter International Synposium on Information and Communication Technologies*. [S.l.]: Trinity College Dublin, 2004. (WISICT '04), p. 1–6.

GRAVES, A.; MOHAMED, A.; HINTON, G. Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.: s.n.], 2013. p. 6645–6649. ISSN 1520-6149.

GROVER, L. K. Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Phys. Rev. Lett.*, v. 79, n. 2, p. 325–328, 1997.

GUPTA, S.; ZIA, R. Quantum neural networks. *Journal of Computer and System Sciences*, v. 63, n. 3, p. 355 – 383, 2001. ISSN 0022-0000. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022000001917696>.

HASSOUN, M. H. *Associative neural memories*. [S.l.]: Oxford University Press, Inc., 1993.

HAYKIN, S. *Neural Networks*. [S.l.]: Prentice Hall, 1999.

HERZ, A. V. M.; GOLLISCH, T.; MACHENS, C. K.; JAEGER, D. Modeling Single-Neuron Dynamics and Computations: A Balance of Detail and Abstraction. *Science*, v. 314, n. 5796, p. 80–85, 2006.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, v. 79, n. 8, p. 2554–2558, 1982.

HOPFIELD, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 81, n. 10, p. 3088–3092, 1984.

HU, W. Towards a real quantum neuron. *Natural Science*, Scientific Research Publishing, v. 10, n. 03, p. 99, 2018.

IBM Quantum Experience. 2018. <http://www.research.ibm.com/quantum>. Accessed: 2018-07-09.

JOHANSSON, J.; NATION, P.; NORI, F. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, Elsevier, v. 183, n. 8, p. 1760–1772, 2012.

JOZSA, R. Entanglement and quantum computation. *arXiv preprint quant-ph/9707034*, 1997.

JOZSA, R.; LINDEN, N. On the role of entanglement in quantum-computational speed-up. In: THE ROYAL SOCIETY. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. [S.l.], 2003. v. 459, n. 2036, p. 2011–2032.

KABIR, A. N. M. E.; UDDIN, A. F. M. N.; ASADUZZAMAN, M.; HASAN, M. F.; HASAN, M. I.; SHAHJAHAN, M. Fusion of chaotic activation functions in training neural network. In: *2012 7th International Conference on Electrical and Computer Engineering*. [S.l.: s.n.], 2012. p. 482–485.

KAK, S. On quantum neural computing. *Information Sciences*, v. 83, n. 3–4, p. 143 – 160, 1995.

KILLORAN, N.; BROMLEY, T. R.; ARRAZOLA, J. M.; SCHULD, M.; QUESADA, N.; LLOYD, S. Continuous-variable quantum neural networks. *arXiv preprint arXiv:1806.06871*, 2018.

KISS, T.; JEX, I.; ALBER, G.; VYMETAL, S. Complex chaos in the conditional dynamics of qubits. *Phys. Rev. A*, v. 74, p. 040301, 2006.

KNILL, E. H. Building quantum computers. *IEEE Information Theory Society News Letter*, v. 58, n. IEEE Information Theory Society News Letter, 2008.

KORN, H.; FAURE, P. Is there chaos in the brain? ii. experimental evidence and related models. *Comptes Rendus Biologies*, v. 326, n. 9, p. 787 – 840, 2003. ISSN 1631-0691. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1631069103002002>.

LAMBERT, N.; CHEN, Y.-N.; CHENG, Y.-C.; LI, C.-M.; CHEN, G.-Y.; NORI, F. Quantum biology. *Nature Physics*, Nature Publishing Group, v. 9, n. 1, p. 10, 2013.

LEE, C.-H.; TENG, C.-C. Identification and control of dynamic systems using recurrent fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, v. 8, n. 4, p. 349–366, Aug 2000. ISSN 1063-6706.

LEPORA, N. F.; VERSCHURE, P.; PRESCOTT, T. J. The state of the art in biomimetics. *Bioinspiration & biomimetics*, IOP Publishing, v. 8, n. 1, p. 013001, 2013.

LEPORATI, A.; FELLONI, S. Three "quantum" algorithms to solve 3-sat. *Theoretical Computer Science*, v. 372, n. 2–3, p. 218 – 241, 2007. ISSN 0304-3975. Membrane Computing.

LEPORATI, A.; ZANDRON, C.; MAURI, G. Simulating the fredkin gate with energy-based p systems. *J. UCS*, v. 10, n. 5, p. 600–619, 2004.

MATSUMOTO, G.; AIHARA, K.; HANYU, Y.; TAKAHASHI, N.; YOSHIZAWA, S.; NAGUMO, J. ichi. Chaos and phase locking in normal squid axons. *Physics Letters A*, v. 123, n. 4, p. 162 – 166, 1987. ISSN 0375-9601. Disponível em: <http://www.sciencedirect.com/science/article/pii/0375960187906967>.

MATSUMOTO, K.; TSUDA, I. Calculation of information flow rate from mutual information. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 21, n. 6, p. 1405, 1988.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Kluwer Academic Publishers, v. 5, n. 4, p. 115–133, 1943. ISSN 0007-4985. Disponível em: <http://dx.doi.org/10.1007/BF02478259>.

MIKOLOV, T.; KARAFIÁT, M.; BURGET, L.; ČERNOCKỲ, J.; KHUDANPUR, S. Recurrent neural network based language model. In: *Eleventh Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2010.

MINSKY, M. L. *Computation: finite and infinite machines*. [S.l.]: Prentice-Hall, Inc., 1967.

MOORE, G. E. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, IEEE, v. 86, n. 1, p. 82–85, 1998.

NETO, F. M. de P.; LUDERMIR, T. B.; OLIVEIRA, W. R. D.; SILVA, A. J. da. Fitting parameters on quantum weightless neuron dynamics. In: *Intelligent Systems, 2015 Brazilian Conference on*. [S.l.: s.n.], 2015. v. 4, p. 169–174.

NETO, F. M. de P.; LUDERMIR, T. B.; OLIVEIRA, W. R. D.; SILVA, A. J. da. Solving np-complete problems using quantum weightless neuron nodes. In: *Intelligent Systems, 2015 Brazilian Conference on*. [S.l.: s.n.], 2015. p. 258–263.

NETO, F. M. de P.; LUDERMIR, T. B.; OLIVEIRA, W. R. de; ; SILVA, A. J. da. Quantum perceptron with dynamic internal memory. In: *2018 International Joint Conference on Neural Networks, IJCNN 2018*. [S.l.]: IEEE, 2018.

NETO, F. M. de P.; OLIVEIRA, W. R. de; LUDERMIR, T. B.; SILVA, A. J. da. Chaos in a quantum neuron: An open system approach. *Neurocomputing*, v. 246, p. 3 – 11, 2017. ISSN 0925-2312.

NETO, F. M. de P.; OLIVEIRA, W. R. de; SILVA, A. J. da; LUDERMIR, T. B. Chaos in Quantum Weightless Neuron Node Dynamics. *Neurocomputing*, v. 183, p. 23–38, 2015.

NETO, F. M. de P.; OLIVEIRA, W. R. de; SILVA, A. J. da; LUDERMIR, T. B. On the entanglement dynamics of the quantum weightless neuron. In: *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2017. p. 175–180.

NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. [S.l.]: Cambridge University Press, 2000.

NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. [S.l.]: Cambridge University Press, 2000.

NIEUWENHUIZEN CLAUDIA POMBO, C. F. A. K. I. A. P. V. p. k. T. M. N. C. P. C. F. A. K. I. A. P. V. p. k. T. M. *Quantum Foundations and Open Quantum Systems: Lecture Notes of the Advanced School*. [S.l.]: World Scientific Publishing Company, 2014. ISBN 9814616729,9789814616720.

NIMBARK, H.; SUKHADIA, R.; KOTAK, P. P. Optimizing architectural properties of artificial neural network using proposed artificial bee colony algorithm. In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.: s.n.], 2014. p. 1285–1289.

NJAFA, J.-P. T.; ENGO, S. N.; WOAFO, P. Quantum associative memory with improved distributed queries. *International Journal of Theoretical Physics*, Springer, v. 52, n. 6, p. 1787–1801, 2013.

OHYA, M.; VOLOVICH, I. V. *Quantum Computing, NP-complete Problems and Chaotic Dynamics*. 1999.

OLIVEIRA, W. R. de; LUDERMIR, T. B. Turing machine simulation by logical neural networks. *Artificial Neural Networks*, NorthHolland Amsterdam, v. 2, p. 663–667, 1992.

OLIVEIRA, W. R. de; SILVA, A. J. da; LUDERMIR, T. B. Vector space weightless neural networks. In: *ESANN 2014*. [S.l.: s.n.], 2014. p. 535–540. ISBN 9782874190957.

PANELLA, M.; MARTINELLI, G. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications*, v. 39, n. 1, p. 61–77, 2011.

PANELLA, M.; MARTINELLI, G. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications*, John Wiley Sons, Ltd., v. 39, n. 1, p. 61–77, 2011. ISSN 1097-007X.

PASEMANN, F. Dynamics of a single model neuron. *International Journal of Bifurcation and Chaos*, World Scientific, v. 3, n. 02, p. 271–278, 1993.

PASEMANN, F. A simple chaotic neuron. *Physica-Section D*, Amsterdam: North-Holland, 1980-, v. 104, n. 2, p. 205–211, 1997.

PAULA, F. M. de. *Quantum Weightless Neuron Dynamics*. Dissertação (Mestrado), 2016.

POLLACK, J. B. Implications of recursive distributed representations. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1989. p. 527–536.

QUANG, T.; KHOA, D.; NAKAGAWA, M. Neural network learning based on chaos. Citeseer, 2007.

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.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958.

RUSSEL, S.; NORVIG, P. Inteligência artificial. *Editora Campus*, p. 26, 2004.

SANTOS, A. C. The ibm quantum computer and the ibm quantum experience. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 39, n. 1, 2017.

SANTOS, P. G. dos; SOUSA, R. S.; ARAUJO, I. C.; SILVA, A. J. da. Quantum enhanced cross-validation for near-optimal neural networks architecture selection. *International Journal of Quantum Information*, World Scientific, p. 1840005, 2018.

SCHULD, M.; FINGERHUTH, M.; PETRUCCIONE, F. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhysics Letters)*, v. 119, n. 6, p. 60002, 2017. Disponível em: <http://stacks.iop.org/0295-5075/119/i=6/a=60002>.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. An introduction to quantum machine learning. v. 56, 09 2014.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. The quest for a quantum neural network. *Quantum Information Processing*, Kluwer Academic Publishers, Hingham, MA, USA, v. 13, n. 11, p. 2567–2586, nov. 2014. ISSN 1570-0755. Disponível em: <http://dx.doi.org/10.1007/s11128-014-0809-8>.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. Simulating a perceptron on a quantum computer. *Physics Letters A*, v. 379, n. 7, p. 660 – 663, 2015. ISSN 0375-9601.

SHOR, P. W. Algorithms for quantum computation: Discrete logarithms and factoring. In: IEEE. *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. [S.l.], 1994. p. 124–134.

SHOR, P. W. Why haven't more quantum algorithms been found? *Journal of the ACM (JACM)*, ACM, v. 50, n. 1, p. 87–90, 2003.

SIEGELMANN, H. T.; SONTAG, E. D. Turing computability with neural nets. *Applied Mathematics Letters*, v. 4, n. 6, p. 77 – 80, 1991. ISSN 0893-9659. Disponível em: <http://www.sciencedirect.com/science/article/pii/089396599190080F>.

SIEGELMANN, H. T.; SONTAG, E. D. On the computational power of neural nets. In: ACM. *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.], 1992. p. 440–449.

SILVA, A. J. da; OLIVEIRA, W. R. de; LUDERMIR, T. B. Classical and superposed learning for quantum weightless neural networks. *Neurocomputing*, v. 75, n. 1, p. 52–60, 2012.

SILVA, A. J. da; OLIVEIRA, W. R. de; LUDERMIR, T. B. Training a classical weightless neural network in a quantum computer. *Quantum*, Citeseer, v. 1, n. 1, p. 1, 2014.

SILVA, A. J. da; OLIVEIRA, W. R. de; LUDERMIR, T. B. Comments on "quantum m-p neural network". *International Journal of Theoretical Physics*, v. 54, n. 6, p. 1878–1881, 2015.

SILVA, A. J. da; OLIVEIRA, W. R. de; LUDERMIR, T. B. Weightless neural network parameters and architecture selection in a quantum computer. *Neurocomputing*, Elsevier, v. 183, p. 13–22, 2016.

SILVA, A. J. da; OLIVEIRA, W. R. de; LUDERMIR, T. B. Weightless neural network parameters and architecture selection in a quantum computer. *Neurocomputing*, v. 183, p. 13 – 22, 2016. ISSN 0925-2312. Weightless Neural Systems.

STROGATZ, S. H. *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity)*. [S.l.]: Westview Press, 1994.

TIAN, J.; LI, M.; CHEN, F.; FENG, N. Learning subspace-based rbfnn using coevolutionary algorithm for complex classification tasks. *IEEE Transactions on Neural Networks and Learning Systems*, v. 27, n. 1, p. 47–61, Jan 2016. ISSN 2162-237X.

TRUGENBERGER, C. Quantum pattern recognition. *Quantum Information Processing*, Kluwer Academic Publishers-Plenum Publishers, v. 1, n. 6, p. 471–493, 2002. ISSN 1570-0755. Disponível em: <http://dx.doi.org/10.1023/A%3A1024022632303>.

TRUGENBERGER, C. A. Quantum Pattern Recognition. *Quantum Information Processing*, v. 1, n. 6, p. 471–493, 2002.

VENTURA, D.; MARTINEZ, T. Initializing the amplitude distribution of a quantum state. *Foundations of Physics Letters*, v. 12, n. 6, p. 547–559, 1999. ISSN 1572-9524.

VENTURA, D.; MARTINEZ, T. Quantum associative memory. *Information Sciences*, v. 124, n. 1–4, p. 273 – 296, 2000. ISSN 0020-0255.

VERSCHURE, P. Chaos-based learning. *Complex Systems*, v. 5, n. S 359, p. 370, 1991.

WEINGARTEN, C. P.; DORAISWAMY, P. M.; FISHER, M. A new spin on neural processing: quantum cognition. *Frontiers in human neuroscience*, Frontiers, v. 10, p. 541, 2016.

WIEBE, N.; KLIUCHNIKOV, V. Floating point representations in quantum circuit synthesis. *New Journal of Physics*, IOP Publishing, v. 15, n. 9, p. 093041, 2013.

XU, B.; WANG, N.; CHEN, T.; LI, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

YANG, D.; LI, G.; CHENG, G. On the efficiency of chaos optimization algorithms for global optimization. *Chaos, Solitons Fractals*, v. 34, n. 4, p. 1366 – 1375, 2007. ISSN 0960-0779. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0960077906004334>.

YEUNG, D. S.; LI, J.; NG, W. W. Y.; CHAN, P. P. K. Mlpnn training via a multiobjective optimization of training error and stochastic sensitivity. *IEEE Transactions on Neural Networks and Learning Systems*, v. 27, n. 5, p. 978–992, May 2016. ISSN 2162-237X.

YU, W. Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms. *Information Sciences*, v. 158, p. 131 – 147, 2004. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025503002032>.

ŻYCZKOWSKI, K. ; HORODECKI, P.; HORODECKI, M.; HORODECKI, R. Dynamics of quantum entanglement. *Phys. Rev. A*, American Physical Society, v. 65, p. 012101, Dec 2001. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevA.65.012101>.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, v. 14, n. 1, p. 35 – 62, 1998. ISSN 0169-2070. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0169207097000447>.

ZHANG, G. P. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 30, n. 4, p. 451–462, Nov 2000. ISSN 1094-6977.

ZHOU, R.; DING, Q. Quantum m-p neural network. *International Journal of Theoretical Physics*, Springer US, v. 46, n. 12, p. 3209–3215, 2007. ISSN 0020-7748.

ZHOU, R.; WANG, H.; WU, Q.; SHI, Y. Quantum associative neural network with nonlinear search algorithm. *International Journal of Theoretical Physics*, v. 51, n. 3, p. 705–723, 2012. ISSN 1572-9575.

ZHOU, R.; WANG, H.; WU, Q.; SHI, Y. Quantum Associative Neural Network with Nonlinear Search Algorithm. *International Journal of Theoretical Physics*, v. 51, n. 3, p. 705–723, 2012.

APÊNDICE A – QUANTUM PERCEPTRON WITH DYNAMIC INTERNAL MEMORY

Quantum Perceptron with Dynamic Internal Memory

Fernando M de Paula Neto and Teresa B Ludermir
Centro de Informática
Universidade Federal de Pernambuco
Recife, Cidade Universitária, 50670-901
Email: {fmpn2, tbl}@cin.ufpe.br

Wilson R de Oliveira and Adenilton J da Silva
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco
Recife, Dois Irmãos, 52171-900
Email: wilson.rosa@gmail.com, adenilton.silva@ufrpe.br

***Abstract*—Motivated by the fact that biological neuron change its content in time and the dynamics of a neuron with internal state can mimic neuronal behaviour such as chaos and bifurcation, neural networks composed of classical artificial neuron models that consider internal information modification after their operation, were introduced in the 90's. Here we study a quantum version of these networks. We introduced the quantum perceptron with internal memory state that can be changed during the neuron execution. For that, we use the quantum perceptron which reproduces the step function of the inner product between input and weights and extend it with a memory that can be updated during its own execution.**

I. Introduction

One of the earliest mathematical approach for modelling the neural cells is the perceptron [1]. For a given n input $x_k \in \{-1, 1\}$, $k = 1, ..., n$ and n weights, $w_k \in [-1, 1)$, the output of the perceptron is controlled by the inner product between the inputs and weights as it is shown in the following equation

$$y = \begin{cases} 1, & \text{if } \sum_{k=1}^{n} w_k x_k \geq 0, \\ -1, & \text{otherwise.} \end{cases} \quad (1)$$

This formula emulates the decision if the neuron propagates an output signal for a given input. A perceptron can be trained by learning algorithms to adjust its weights to classify linearly separable problems. With combination of several layers of perceptrons it is possible to classify non-linearly separable problems and to approximate rich set of practical functions [2].

Quantum computation (**QC**) has shown some advantages over classical computation. It is possible with **QC** to solve the factoring problem in polynomial time [3] and to search in an unordered database quadratically faster than any existing classical algorithm [4]. The quantum parallelism which allows to compute many values of a function in a single execution and *entanglement* are intrinsically versatile features of **QC**.

Quantum systems are considered to have atypical patterns and it is often taken as non-intuitive and sometimes even paradoxical. By observing the behaviour of quantum systems, it is expected that quantum algorithms could recognize complex statistical patterns and intrinsic data information [5]. The nonlocality feature of quantum theory appears to be in accordance with some brain functions [6]. For all these reasons quantum algorithms for machine learning tasks have been proposed to solve real problems, of course the full advantage being realised only when actual a quantum computer will be built. The works which lead with quantum computing and machine learning have grown to propose new methods joining these areas [5].

In [7], it was shown that biological neurons have interesting properties during their dynamics such as chaos and dynamic internal memory. Experiments with squid giant axons and numerically with the Hodgkin-Huxley equations show that neurons responses are not always periodic and that the apparently nonperiodic responses can be understood as deterministic chaos [8]. It is then desirable that artificial neurons may also have those properties. In this paper, our main contribution is to discuss a model of quantum neuron which has a dynamic internal memory. Previous works did not have this feature. qRAM memory trains its selectors and uses them to recover a content in the memory. The modification of the memory after the qRAM reading is not considered [9]. In [10], a quantum associative memory uses an adaptation of the Grover's algorithm [4]. This model proposes a pattern storing algorithm [11] which creates a quantum state from a training set for patterns of length n bits using $2n + 1$ quantum bits. In the retrieval processing, Grover's algorithm is adapted and the memory content can be recovered. In this model, the dynamics of the memory is not considered. Only the recovery procedure is detailed. In [12], the quantum storing pattern algorithm uses $2n + 2$ qubits and executes a unitary retrieval through the Hamiltonian operator which calculates the *Hamming distance* between two binary patterns. In this retrieval algorithm, it is considered the dynamics of the Hamiltonian system. The authors argue that this Hamiltonian is the generalisation of the

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

Hopfield model with efficiently find the exact global minimum of the quantum energy landscape, without the appearance of any spurious memories (*i.e* patterns not desired but possibly included during the training step). Even though the dynamics of the memory during the recovery procedure is considered, the memory after this recovery is not. The authors propose to use some correction algorithm to not lost completely the memory after the proposed procedure. Other models do not consider the dynamical aspect of the memory [13], [14], [15], [16].

Aiming to consider the internal memory dynamics we adapt an existing quantum perceptron model proposed by [17] adding the internal state as a memory. We discuss how the modification of the internal state can occur and we show some cases in which its dynamical effects can modify the computational ability of the neuron. In Section II, a brief review of quantum computation is done with concepts used in this paper. In Section III, quantum neurons proposed before are discussed and in Section IV the proposed model is presented. The conclusions and final considerations are in Section V.

II. Quantum computation

A. Quantum bits

The unit of information in quantum computation is called a *quantum bit* (qubit). A qubit is a unit vector in a two-dimensional complex vector space $\mathbb{C}^2$. It can be in superposition of the *basic states*, *i.e.* in the position 0 or in the position 1 at the same time, if we consider the canonical basis as $|0\rangle$ and $|1\rangle$. Any qubit $|\psi\rangle$, as a vector (or *state*) of $\mathbb{C}^2$, can be written as a linear combination of the canonical (or *computational*) basis $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$ as viewed in Equation (2),

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2}$$

where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. This notation also means that upon measuring the qubit the result is $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$.

The qubits are represented mathematically together by the tensor operator, $\otimes$. The tensor operator is used to represent quantum systems with two or more qubits $|\mathbf{g}\rangle = |ij\rangle = |i\rangle \otimes |j\rangle$. Here we will use the bold font for the representation of quantum states with more one qubit. For two qubits $|i\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|j\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, the tensor operator generates the state $|ij\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle$. For a general two states $|\mathbf{p}\rangle$ and $|\mathbf{q}\rangle$ with n and m states respectively, the state $|pq\rangle$ can be calculated by the operation described in Equation 3.

$$\begin{bmatrix} \alpha_1 \\ \dots \\ \alpha_{2^n} \end{bmatrix} \otimes \begin{bmatrix} \beta_1 \\ \dots \\ \beta_{2^m} \end{bmatrix} = \begin{bmatrix} \alpha_1 \begin{bmatrix} \beta_1 \\ \dots \\ \beta_{2^m} \end{bmatrix} \\ \dots \\ \alpha_{2^n} \begin{bmatrix} \beta_1 \\ \dots \\ \beta_{2^m} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \alpha_1\beta_1 \\ \dots \\ \alpha_1\beta_{2^m} \\ \dots \\ \alpha_{2^n}\beta_1 \\ \dots \\ \alpha_{2^n}\beta_{2^m} \end{bmatrix} \tag{3}$$

We can represent the quantum states using integer numbers rather than string bits inside the $|\cdot\rangle$ notation. For a given quantum state with n states $|\boldsymbol{\psi}\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdots + \alpha_n|n\rangle$ the measurement of the $|x\rangle$ state may occur with $|\langle\boldsymbol{\psi}||x\rangle|^2$ of probability where the $\langle\cdot|$ represents the complex conjugate of the vector $|\cdot\rangle$.

Let Q and R be two vector spaces the tensor product of Q and R, denoted by $Q \otimes R$, is the vector space generated by the tensor product of all vectors $|a\rangle \otimes |b\rangle$, with $|a\rangle \in A$ and $|b\rangle \in B$. Some states $|\psi\rangle \in Q \otimes R$ cannot be written as a product of states of its component systems Q and R. States with this property are called *entangled* states. For instance, two entangled qubits are the Bell states described in Equation (4).

$$\begin{aligned} |\Phi^+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad |\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ |\Phi^-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \qquad |\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \tag{4}$$

B. Quantum operators

The quantum states are modified by quantum operators that change the amplitude values of the qubits. A quantum operator $\mathbf{U}$ over n qubits is a unitary complex matrix of order $2^n \times 2^n$. For instance, some operators over 1 qubit are: Identity (**I**), Not (**X**) and Hadamard (**H**), described below in Equation (5) and Equation (6) in matrix representation and their working in the computational basis. The combination of unitary operators forms a quantum circuit.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{aligned} \mathbf{I}|0\rangle &= |0\rangle \\ \mathbf{I}|1\rangle &= |1\rangle \end{aligned} \quad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{aligned} \mathbf{X}|0\rangle &= |1\rangle \\ \mathbf{X}|1\rangle &= |0\rangle \end{aligned} \tag{5}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{aligned} \mathbf{H}|0\rangle &= 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle &= 1/\sqrt{2}(|0\rangle - |1\rangle) \end{aligned} \tag{6}$$

The Identity operator **I** generates the output exactly as the input; **X** operator works as the classic NOT in the computational basis; Hadamard **H** generates a superposition of states when applied in a computational basis.

In the same way we can combine quantum states, quantum operators can also be combined using tensor product. For two (n_0, m_0)-dimensional matrix U and (n_1, m_1)-dimensional matrix V, their composition, $U \otimes V$, products a third $(n_0 n_1, m_0 m_1)$-dimensional matrix. We denote as $\mathbf{A}^{\otimes s}$ the s-fold application of $\mathbf{A}$.

The **CNOT** is a two qubits operator. It has a control qubit and a target qubit. It works considering the value of the control qubit to apply the **X** operator on the target qubit. If the control qubit is set to 1 the **X** operator is applied to target qubit. The matrix representation in the computational basis is shown in Equation 7.

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{array}{l} \mathbf{CNOT}|00\rangle = |00\rangle \\ \mathbf{CNOT}|01\rangle = |01\rangle \\ \mathbf{CNOT}|10\rangle = |11\rangle \\ \mathbf{CNOT}|11\rangle = |10\rangle \end{array} \tag{7}$$

C. Quantum circuit

We can represent quantum operations by quantum circuits. This graphical representation considers the qubits as wires and quantum operators as boxes. The flow of the execution, as in the classical case, is from left to right.

Figure 1 has an example of a quantum circuit composed of a **CNOT**, where the control qubit is depicted by a filled circle and the symbol $\oplus$ indicating the target qubit, and a **X** operator.

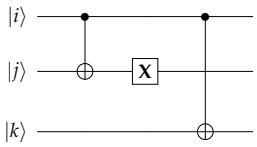


Fig. 1: An example of quantum circuit with two **CNOT** operators and a **X** operator.

In quantum computing, almost all the computation is unitary. Then the operations in these cases must be reversible. Hence if a non-invertible function is implemented, the original inputs need to appear in the output to be possible to recover the information in a reversible way. For a given function f, considering that the operator $\mathbf{U}_f$ implements this function, it is common to encounter the quantum circuit representation of this function as it is shown in Figure 2.

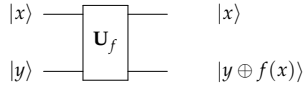


Fig. 2: Quantum circuit to represent an operation of an arbitrary boolean function f implemented by the quantum operator $\mathbf{U}_f$, where the symbol $\oplus$ is the XOR, the exclusive OR-operation (binary addition modulo 2). The result of the application of $|x\rangle$ in the function will be appear in the second quantum state when $|y\rangle = |0\rangle$ since $|y \oplus f(x)\rangle = |0 \oplus f(x)\rangle = |f(x)\rangle$.

D. Quantum Fourier Transform

The problem of finding prime factors of an integer could be efficiently built in a quantum computer by the Shor algorithm implementation [3]. This algorithm takes account the phase estimation of a quantum state. In this Section, we present the Quantum Fourier Transform (QFT) which is a tool widely used in quantum algorithms and important in the step of quantum phase estimation.

The QFT is a linear transformation on n qubits for a given computational basis state $|j\rangle$, where $0 \leq j \leq 2^n - 1$. This procedure is shown in Equation 8.

$$|j_1 j_2 ... j_n\rangle = |j\rangle \rightarrow |j'\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle \tag{8}$$

The QFT means that we can represent a given quantum state $|j\rangle$ in a new quantum state $|j'\rangle$ where its phase stores the information of the original quantum state $|j\rangle$. As the QFT is a linear operator, its inverse operator can recovery the information in the phase of a quantum state, transforming it in a respective quantum state.

III. Quantum neurons

There are some contributions to map the behaviour of the neuron in the quantum computation perspective. The main challenge is that the quantum operations are unitary and the neuron carries a nonlinear operator in this working. The parallelism and entanglement are also expected to be used as an advantage since these features are presented intrinsically as a quantum tool [18], [19].

In [13], a description of a quantum neural network model is given. In this model, the neural network saves its information in a quantum operator. For a given quantum system with n inputs $|x_1\rangle \dots |x_n\rangle$, the output of this neuron is $|y\rangle = \hat{F} \sum_{j=1}^{n} \hat{w}_j |x_j\rangle$ where $\hat{w}$ is a 2x2 matrix operator which saves the neuron information. The training of this $\hat{w}$ quantum operator is done by an iterative process described in Equation 9 where $|d\rangle$ is the desired state.

$$\hat{w}_j(t+1) = \hat{w}_j(t) + \eta(|d\rangle - |y(t)\rangle)\langle x_j| \tag{9}$$

In [20], it was demonstrated that the learning rule described in [13] does not preserve the unitary of the operators. A quantum neuron whose weights are into a quantum operator was also proposed in [21]. In this model, the learning algorithm adjusts the weights of the neuron according the expected output. This operation is described in Equation 10.

$$w_{ij}^{t+1} = w_{ij}^{t} + \eta(|O\rangle_i - |\psi\rangle_i)|\phi\rangle_j \tag{10}$$

The w_{ij} are the matrix entries indexed by row i and column j; η is the learning rate, $|\phi\rangle$ and $|O\rangle$ are respectively examples of input and expected output in the training set, W is the weight matrix of the neuron and

$|\psi\rangle$ the application of the input in the weight matrix $|\psi\rangle = W^t|\phi\rangle$. It was shown in [22] that this neuron can be efficiently simulated in a classical neural networks.

In [15], it was proposed to use a nonlinear operator to find the best parameters of a neural network in a superposition way. The quantum architecture allows to use the parallelism to evaluate all the possible weights of the networks and the nonlinear proposed in [23] makes an exhaustive search of the optimal parameters. In [24] and [25], the quantum RAM based neuron was defined as the quantisation of the weightless neural networks proposed in [26]. The RAM node stores in its memory one bit addressed by an input bit string. The qRAM represents that bit storage by the gate A, that is a **CNOT** gate. A quantum neuron viewed over a perspective of the time evolution of a single quantum object is demonstrated in [14]. The neuron as a memory which recoveries the information based on Grover's search algorithm is proposed in [16].

In [17], a model of the quantum perceptron is proposed. This model simulates the classical perceptron which has the step function as neuron activation function. It uses the inverse of the quantum Fourier transform algorithm to calculate the inner product between the input and the weights of the neuron. The step function is simulated when the first qubit of the inner product result is measured. If this qubit is 1, with some probability, the inner product is more than $1/2$. The probability of success depends on the precision of qubits we use to represent the inner product. In Figure 3, it is shown the quantum circuit which calculates the inner product with the weights w and the input $|x\rangle$. In this model, the weights are fixed in quantum operators. The algorithm starts with some τ zeroes, in which the τ is the precision used, and the input $|\psi_0\rangle = |x_1, x_2, ..., x_n\rangle$. The operator $U_k(w_k)$, described in Equation 11, where $\Delta_\phi = \frac{1}{2n}$, applies the phase change by kth input register qubit.

$$U_k(w_k) = \begin{pmatrix} e^{-2\pi i w_k \Delta_\phi} & 0 \\ 0 & e^{2\pi i w_k \Delta_\phi} \end{pmatrix} \tag{11}$$

Then the inverse quantum Fourier transform calculates the phase change included in the input state through the $U(\hat{w}) = U_n(w_n)... \otimes ...U_2(w_2) \otimes U_1(w_1)$ operator. $|J\rangle = |J_1, ..., J_\tau\rangle$ is the binary representation of the integer j and $\phi = \frac{j}{2^\tau}$. For the state be in the way that we can calculate the estimation of the inner product of the input with the weight vector, the algorithm applies sometimes the controlled-$U(\hat{w})^{2^j}$ operation (called *modular exponentiation*) to get the result quantum state in the shape where the inverse quantum Fourier transform can calculate the phase estimation correctly. The modular exponentiation can be done using the trick of uncomputation and details of implementation are encountered in [19]. In Equation 12 the operation of quantum inverse Fourier is shown where the phase ϕ is estimated as $\hat{\phi}$.

$$\frac{1}{\sqrt{2^\tau}} \sum_{j=0}^{2^\tau - 1} \exp^{2\pi i j \phi} |J\rangle |\psi_0\rangle \xrightarrow{QFT^{-1}} |\hat{\phi}\rangle |\psi_0\rangle \tag{12}$$

Schuld and collaborators [17] also introduce a slight variation of this quantum perceptron including the weights as quantum registers. This allows to put the weights also as an input of the circuit. The initial state considers the input and the weights of the neuron

$$\left| x_1, ..., x_n; W_1^{(1)}, ..., W_1^{(\delta)}, ..., W_n^{(1)}, ..., W_n^{(\delta)} \right\rangle = |\mathbf{x}; \mathbf{w}\rangle \tag{13}$$

$W_k^{(m)}$ is the mth digit of the binary fraction representation that express w_k as $w_k = W_k^1 \frac{1}{2} + ... + W_k^{(\delta)} \frac{1}{2^\delta}$ with a precision δ. For this modification, it is introduced the controlled two-qbuit operator

$$U_{w_k^{(m)}, x_k} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-2\pi i \Delta_\phi \frac{1}{2^m}} & 0 \\ 0 & 0 & 0 & e^{2\pi i \Delta_\phi \frac{1}{2^m}} \end{pmatrix} \tag{14}$$

The mth bit $W_k^{(m)}$ of the binary representation of w_k controls the operation of shifting the phase by $\Delta\phi \frac{1}{2^m}$ (for $x_k = 0$) or $\Delta\phi \frac{1}{2^m}$ (for $x_k = 1$), using $\Delta\phi$ from above. For more details, see [17].

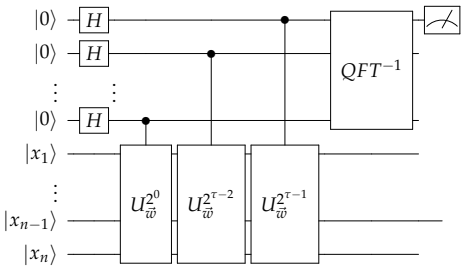


Fig. 3: Quantum perceptron circuit proposed in [17].

In [6], the brain is analysed as a quantum system. This model points out that the neuron can include an internal dynamics. It considers the process of recognition changes the internal content of the neuron. This internal modification is viewed as a memory space which is dynamical. In [7], it is shown that internal state in a neuron induces a chaotic dynamics and simulates more realistic the working of the biologic neuron. In the following section, we introduce a model of a quantum neuron which has an internal dynamical memory which was not discussed before. We show how the behaviour of the neuron is increased since the internal content is considered and we give some considerations about the training of this neuron model.

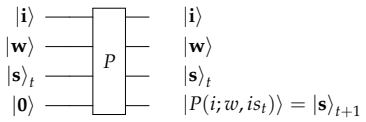


Fig. 4: Quantum circuit of the quantum perceptron of type A. This model considers that the internal state of the neuron is the neuron output in the last iteration.

IV. Quantum Perceptron Models with internal memory

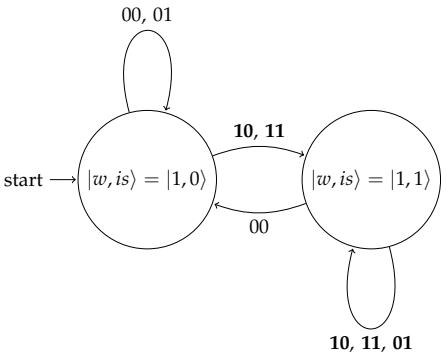
In this section, we propose to discuss the internal memory modification as a feature of a quantum perceptron. Then we modify the quantum perceptron proposed in [17] and we discuss how this modification affects the functioning of the neuron. The quantum perceptron of [17] is considered as a black box named as P (we explain its general working in the Section III). In this case, the perceptron has inputs $|\mathbf{i}\rangle$ and weight memories $|\mathbf{w}\rangle$ and internal state $|\mathbf{s}\rangle$, which are separated only to stay more clear during our operations, since the $|\mathbf{s}\rangle$ can be modified after neuron iteration. Initially it can be viewed as only one weight vector $|\mathbf{w},\mathbf{is}\rangle$. In our neuron operation we consider that after the application of the P operator, the input qubits can be recovered by reversible computing techniques [19].

We propose three approaches to represent the quantum perceptron with internal memory (**QPIM**). In the first case, which we name **QPIM** type A, the internal state memory is changed by the output generated by the perceptron. In the second case, called **QPIM** type B, the internal state can be changed by unitary operators without auxiliary qubits. In that case, the functions that modify the internal memory are characterised to be only the reversible functions. And, in the third case, **QPIM** type C, the internal state is modified by an arbitrary non-reversible boolean function via the $\mathbf{U}_f$ operator.

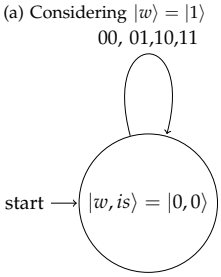
A. Quantum perceptron with internal memory type A

In this model, the quantum perceptron P calculates the inner product between $|i\rangle$ and $|w,is\rangle$ and calculates the step function measuring the first qubit of the output operation. This output qubit is now the new internal state $|\mathbf{s}\rangle$ of the neuron. The circuit representation of this model is shown in Figure 4.

With the inclusion of the internal state dynamics, the neuron can consider, in its calculus, the neuron result in the previous iteration. The neuron works like an automaton that has low memory but it can memorise the current state of its execution. In Figure 5, automatons are shown considering the weight $|w\rangle = |0\rangle$ and $|w\rangle = |1\rangle$ and internal state starting as $|0\rangle$. Analysing these automata, it is simple to see that for a given input, *e.g* $|01\rangle$, the neuron output will depend of the neuron current state.



(a) Considering $|w\rangle = |1\rangle$



(b) Considering $|w\rangle = |0\rangle$

Fig. 5: Automatons representing the dynamics of the Quantum Perceptron of type A in the two cases of the values of weight $|0\rangle$ and $|1\rangle$. The internal state is considered to start with $|0\rangle$ and the excitation threshold is $\theta = 1$.

The neuron with internal memory is not a static function. The inputs which are bold are the inputs that excite the neuron, *i.e* the output is 1. We can see that for different values of weights the working of the neuron is modified. In this model, it is dependent that the inner product between input and weights can be more than the threshold to change the internal state. If it is not possible, the neuron can be in only one possible state.

B. Quantum perceptron with internal memory type B

In this model, after the inner product calculus, the quantum perceptron modifies the content of the internal state considering in its modification any reversible function operation. In other words, the internal state is modified by any reversible function. Figure 6 shows the quantum circuit of this model. Since no ancillary qubits are used to perform the internal state modification, only reversible function can be executed to alter the internal state. The U_B operator can be any composition of unitary

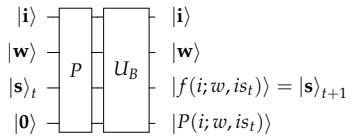


Fig. 6: Quantum circuit of the quantum perceptron of type B. This neuron model has the internal state modified by a reversible function in function of the input, weights and internal state in the previous iteration.

operators with variables $|i\rangle$, $|w\rangle$ and the current internal state $|\mathbf{s}\rangle$.

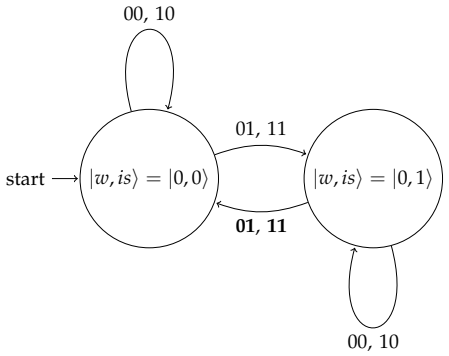
In this neuron model, one increases the amount of functions to be possible to alter the internal state. Not only the neuron activation function can be considered to alter it. Temporal information or specific configuration of inputs and weights can modify the internal memory state to some specific values. Figure 7 shows automatons with different values of weights and consider all possible input values in the state transitions. The U_B operator of this neuron is the $CNOT$ gate with the second qubit of the input being the control qubit and the internal state qubit as the target. The bold inputs are the ones that excite the neuron, *i.e.* the output is 1. In these automatons we can see the same inputs exciting or not the neuron depending of the current neuron state.

C. Perceptron with internal memory type C

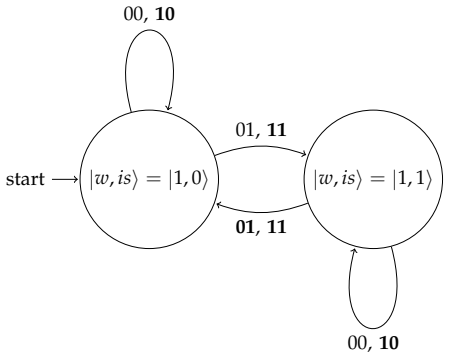
The quantum perceptron with internal memory which we name type C considers the possibility to modify the internal state by an arbitrary boolean function. This is possible because we are considering ancillary qubits to help in this computation. In Figure 8, the quantum circuit of this model is shown. The function, that alters the content of the internal state of the neuron, is represented by U_C operator.

Examples of automatons leading with the dynamics of the perceptron of type C is shown in Figure 9. In this example, the U_C modifies the internal state when the two last qubits of the input are equal to internal state qubits. The modification is to flip all the qubits of the internal state. The values of input which excite the neuron are in bold font. The working of the neuron is completely dependent on the weight but also on the internal state of the last iteration.

The three quantum neuron models presented with internal memory have a limited number of qubits as internal state. Type C is the model which requires more qubits due to the ancillary cubits needed for simulating an arbitrary boolean function not necessarily reversible one. One can see that the modification of internal state by the output of the neuron (neuron model type A), by reversible functions (neuron model type B) and by an



(a) Considering $|w\rangle = |0\rangle$.



(b) Considering $|w\rangle = |1\rangle$.

Fig. 7: Automatons representing the dynamics of the Quantum Perceptron of type B in the two cases of the values of weight $|0\rangle$ and $|1\rangle$. The internal state is considered to start with $|0\rangle$ and the excitation threshold is $\theta = 1$. We see in this automatons that the neuron is excited by different inputs in dependence of the weight qubit. The internal state

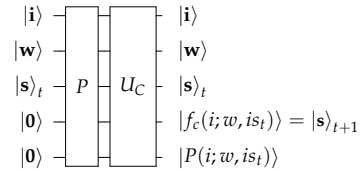
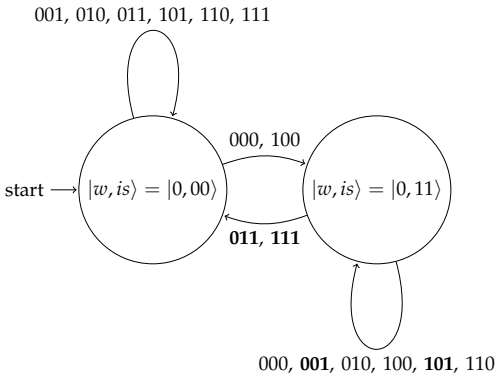
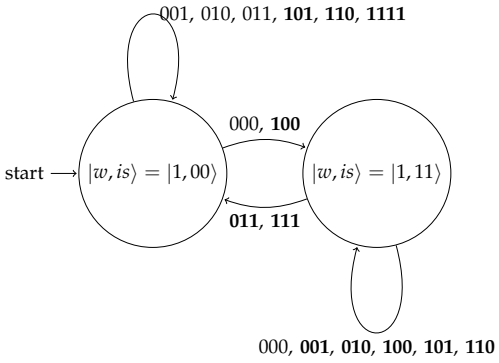


Fig. 8: Quantum circuit of the quantum perceptron of type C. This neuron mode has the internal state modified by an arbitrary boolean function.



(a) Considering $|w\rangle = |0\rangle$.



(b) Considering $|w\rangle = |1\rangle$.

Fig. 9: Automatons representing the dynamics of the Quantum Perceptron of type C considering the two cases of the values of weight $|0\rangle$ and $|1\rangle$. The internal state is considered to start with $|0\rangle$ and the excitation threshold is $\theta = 1$.

arbitrary boolean function (neuron model type C), allow enriching the dynamics of the neuron. The model type A depends to change the internal state of the input and of the weights to exceed the threshold. This dependence generates a dynamics with a single state if all possible inputs could not excite the neuron. This example case can be viewed in Figure 5. The second model type B allows changing the content of the internal state neuron with reversible functions. This is considered in case that the neuron has no ancillary or auxiliary qubits to compute its functions. In other words, the memory is limited by the qubits of the inputs. In this case, we can increase the number of states because the internal state can be modified when either input or weights are in an expected configuration. In neuron model type C, the internal modification is defined vast to any function modification. This is can be possible when even the input is not enough to excite the neuron, the internal state is modified. There is, in this case, some available memory.

V. Conclusion

In this paper, we introduced the quantum perceptron with internal state memory that can be changed during the neuron execution. For that, we use the quantum perceptron which reproduces the step function of the inner product between input and weights proposed by [17] and put a memory that can be updated during its own execution. The relevance of this approach is because some previous works point out that biologic neuron changes its content in time [6] and the dynamics of this neuron with an internal state can mimic some biological neuron behaviours such as chaos and bifurcation [27]. Hence we introduced some quantum neuron models which change in different ways the content of the internal state according to quantum computation. It is possible to perceive that the neuron output dependent on the internal state. Then the perceptron is not a static function but a dynamics processor which adapts itself in each iteration.

In all the proposed models, the neuron output is governed by the step function of the perceptron. Future works intend to generalise the activation function of the neuron, using the quantum computation to perform optimized functions. To train a quantum perceptron with internal memory network are also considered to solve real problems, mainly those which has time dependence.

Acknowledgment

The authors would like to thank CNPq and FACEPE (Brazilian Research Agencies) for their financial support.

References

[1] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, pp. 65–386, 1958.

[2] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, Nov 2000.

[3] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[4] L. K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997.

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195 EP –, Sep 2017.

[6] K.-H. Fichtner, K. Inoue, and M. Ohya, "On a quantum model of brain activities — distribution of the outcomes of eeg-measurements and random point fields," *Open Systems & Information Dynamics*, vol. 19, no. 04, p. 1250025, 2012.

[7] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," *Physics Letters A*, vol. 144, no. 6, pp. 333 – 340, 1990.

[8] Y. Hirata, M. Oku, and K. Aihara, "Chaos in neurons and its application: Perspective of chaos engineering," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 047511, 2012.

[9] W. R. de Oliveira, A. J. da Silva, and T. B. Ludermir, "Vector space weightless neural networks," in *ESANN 2014*, no. April, 2014, pp. 535–540.

[10] D. Ventura and T. Martinez, "Quantum associative memory," *Information Sciences*, vol. 124, no. 1–4, pp. 273 – 296, 2000.

[11] ——, "Initializing the amplitude distribution of a quantum state," *Foundations of Physics Letters*, vol. 12, no. 6, pp. 547–559, 1999.

[12] C. A. Trugenberger, "Probabilistic Quantum Memories," *Phys. Rev. Lett.*, vol. 87, no. 6, p. 67901, 2001.

[13] M. V. Altaisky, "Quantum neural network," Joint Institute for Nuclear Research, Russia, Technical report, 2001.

[14] E. Behrman, L. Nash, J. Steck, V. Chandrashekar, and S. Skinner, "Simulations of quantum neural networks," *Information Sciences*, vol. 128, no. 3–4, pp. 257 – 269, 2000.

[15] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *International Journal of Circuit Theory and Applications*, vol. 39, no. 1, pp. 61–77, 2011.

[16] B. Ricks and D. Ventura, "Training a quantum neural network," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, 2004, pp. 1019–1026.

[17] M. Schuld, I. Sinayskiy, and F. Petruccione, "Simulating a perceptron on a quantum computer," *Physics Letters A*, vol. 379, no. 7, pp. 660 – 663, 2015.

[18] R. Jozsa and N. Linden, "On the role of entanglement in quantum-computational speed-up," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 459, no. 2036. The Royal Society, 2003, pp. 2011–2032.

[19] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[20] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir, "Classical and superposed learning for quantum weightless neural networks," *Neurocomputing*, vol. 75, no. 1, pp. 52–60, 2012.

[21] R. Zhou and Q. Ding, "Quantum m-p neural network," *International Journal of Theoretical Physics*, vol. 46, no. 12, pp. 3209–3215, 2007.

[22] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir, "Comments on "quantum m-p neural network"," *International Journal of Theoretical Physics*, vol. 54, no. 6, pp. 1878–1881, 2015.

[23] D. S. Abrams and S. Lloyd, "Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems," *Phys. Rev. Lett.*, vol. 81, no. 18, pp. 3992–3995, 1998.

[24] W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, A. Leonel, W. R. Galindo, and J. C. Pereira, "Quantum Logical Neural Networks," in *Brazilian Symposium on Neural Networks*, 2008, pp. 147–152.

[25] W. R. de Oliveira, "Quantum RAM Based Neural Networks," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 22–24, 2009.

[26] I. Aleksander, "Self-adaptive universal logic circuits," *Electronics Letters 2*, vol. 8, pp. 321–322, 1966.

[27] F. Pasemann, "Dynamics of a single model neuron," *International Journal of Bifurcation and Chaos*, vol. 3, no. 02, pp. 271–278, 1993.

APÊNDICE B – IMPLEMENTING NONLINEAR ACTIVATION FUNCTIONS IN A QUANTUM NEURON

Implementing nonlinear activation functions in a quantum neuron

Fernando M de Paula Neto, *Member, IEEE,* Teresa Bernarda Ludermir, *Senior Member, IEEE,* Wilson R de Oliveira, and Adenilton J da Silva

***Abstract*—The ability of artificial neural networks (ANNs) to adapt to input data and perform generalisations are intimately connected to the use of nonlinear *activation functions*. Quantum versions of ANN have been proposed to take advantage of the possible supremacy of quantum over classical computing. To date, all proposals faced the difficulty of implementing nonlinear activation functions since quantum operators are linear. This paper presents an architecture to simulate the computation of an arbitrary nonlinear function as a quantum circuit. This computation is performed on the phase of an adequately designed quantum state, and quantum phase estimation recovers the result, given a fixed precision.**

***Index Terms*—Quantum neuron, Quantum neural networks, Quantum Fourier transform, Quantum computing**

I. Introduction

THE artificial neuron operates in two stages. The first one is the interaction between inputs and internal weights of the neuron. The function which produces this interaction is called *propagation function*. In the second stage the *activation function* acts on the result of the propagation function producing the output of the neuron [1]. Equation 1 shows f and g, respectively propagation and activation functions of the classical neuron N.

$$N(i_1,...,i_k,w_1,...w_k) = g(f(i_1,...,i_k,w_1,...,w_k)) \quad (1)$$

Quantum computing is providing new tools for processing information [2] which seems to supersede their classical counterparts in efficiency. In particular, there exist a growing effort to design quantum machine learning algorithms [3], [4]. One challenge in quantum neural networks is that the quantum operations in the circuit model are unitary, but the artificial neuron uses nonlinear functions [5].

There are many proposals of quantum neural networks implementing the artificial neuron in quantum circuit model [5]. In [6], dissipative operators implement the intrinsic nonlinearity of an artificial neural network. RAM-based neurons store content in quantum registers and can nonlinearly map the input into the output simulating a RAM (Random Access Memory) [7], [8]. Panella and Martinelli [9] proposes a quantum neural network that implements the nonlinearity of the activation function using suitable Boolean functions that can be simulated by unitary quantum operators, but the learning operator is nonunitary. Some models consider the neuron weights as a unitary operator, but the learning updating is also a nonunitary operator [10], [11], [12]. An exciting model considers the dynamic evolution of quantum dots molecules as executing neural networks [13]. A quantum perceptron model is proposed in [14]. This model simulates a classical perceptron which has the step function as the activation function. In [15], a quantum neuron with a periodic tangent functions and step activation function was implemented by a probabilistic circuit of the class *repeat-until-success (RUS) circuits* [16]. The non-periodicity activation function for this neuron appears in [17]. A quantum neural network based in continuous-variable quantum architecture was proposed in [18]. This model uses another model of quantum computation, and the range of their parameters are continuous. Other neural networks models are quantum-based and do not follow completely the theory of quantum computing [19], [20]. For an updated review of the many quantum neural networks models see [21].

Existing contributions are related to some internal function of quantum neuron models. In this paper, our main contribution is to discuss a general model of a quantum neuron with any activation and propagation functions. The computation of the neuron information propagation is performed in the phase of a quantum state processed by quantum inverse Fourier transform. The inputs and weights of the neuron are quantum registers of a quantum circuit, and the neuron is a quantum operator parameterised by its precision and internal functions. The proposed neuron has properties of classical neurons, and uses quantum features such as superposition and entangled inputs.

II. Quantum computation

A. Quantum bits

A *quantum bit* (qubit) is a unit of information in quantum computation - the quantum analogue of a binary bit. A qubit is a two-state quantum mechanical system, such as the spin of a spin-$\frac{1}{2}$ particle such as an electron, whose spin can have values $+\hbar/2$ or $-\hbar/2$, where $\hbar$ is the

Fernando M Paula Neto and Teresa Bernarda Ludermir are with Centro de Informática, CIn, Universidade Federal de Pernambuco, Brazil, e-mail: fmpn2, tbl@cin.ufpe.br.

Wilson R de Oliveira is with Departamento de Estatística e Informática, DEINFO, and Adenilton J da Silva is with Departamento de Computação, Universidade Federal Rural de Pernambuco, Brazil, e-mail: wilson.rosa@gmail.com and adenilton.silva@ufrpe.br.

Manuscript received X, X; revised X X, X.

reduced Planck constant. A qubit is represented as a two-dimensional vector in the complex vector space $\mathbf{C}^2$. The two independent (physically distinguishable) quantum states of the system are represented as basis states (or basis vectors) in the canonical (or computational) basis as $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$. As such any qubit can be seen as the linear combination (usually called *superposition*) of the basis vectors, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers. Qubits are normalised, which requires that $|\alpha|^2 + |\beta|^2 = 1$. This notation also means that the qubit has probability $|\alpha|^2$ to be measured as 0 and $|\beta|^2$ to be measured as 1 and α and β are then called *probability amplitudes* or simply *amplitudes*.

The tensor operator $\otimes$ is used to represent quantum systems composed of two or more qubits $|\mathbf{g}\rangle = |ij\rangle = |i\rangle \otimes |j\rangle$. Here we will use the bold font for the representation of quantum states with more than one qubit. For two qubits $|i\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|j\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, the tensor operator generates the state $|ij\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle$.

For general two vectors $|\mathbf{p}\rangle$ and $|\mathbf{q}\rangle$, respectively living in a n and m-dimensional vector space, their tensor product $|\mathbf{pq}\rangle = |\mathbf{p}\rangle \otimes |\mathbf{q}\rangle$ can be calculated as described in Equation 2 and live in nm-dimensional vector space.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ ... \\ \alpha_n \end{bmatrix} \otimes \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_m \end{bmatrix} = \begin{bmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_2 \\ ... \\ \alpha_1\beta_m \\ ... \\ \alpha_n\beta_1 \\ \alpha_n\beta_2 \\ ... \\ \alpha_n\beta_m \end{bmatrix} \tag{2}$$

We can represent the quantum states using integer numbers rather than string of bits inside the $|.\rangle$ notation. For a given n-dimensional quantum state the representation can be $|\boldsymbol{\psi}\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdots + \alpha_n|n\rangle$. The $\langle .|$ notation represents the complex conjugate of the vector $|.\rangle$. Let Q and R be two vector spaces, their tensor product denoted by $Q \otimes R$, is the vector space generated by the tensor product of all vectors $|a\rangle \otimes |b\rangle$, with $|a\rangle \in A$ and $|b\rangle \in B$.

B. Quantum operators

Quantum states are modified by quantum operators which change the amplitude values of the qubits. A quantum operator $\mathbf{U}$ acting over a n qubits system is a unitary complex matrix of order $2^n \times 2^n$. A matrix U is unitary if $U \circ U^\dagger = U^\dagger \circ U = I$, where $U^\dagger$ is the adjoint (conjugate transpose) of U, I is the identity matrix and $\circ$, usually omitted, is the usual matrix product.

Some operators over one qubit are: Identity ($\mathbf{I}$), NOT ($\mathbf{X}$) and Hadamard ($\mathbf{H}$) operators, defined in the basis states as $\mathbf{I}|0\rangle = |0\rangle$, $\mathbf{I}|1\rangle = |1\rangle$, $\mathbf{X}|0\rangle = |1\rangle$, $\mathbf{X}|1\rangle = |0\rangle$, $\mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$ and $\mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle)$.

In the same way we can combine quantum states, quantum operators can also be combined using tensor product. For two (n_0, m_0)-dimensional matrix U and (n_1, m_1)-dimensional matrix V, their tensor product, $U \otimes V$, is a $(n_0 n_1, m_0 m_1)$-dimensional matrix. We denote as $\mathbf{A}^{\otimes s}$ the s-fold application of $\mathbf{A}$.

The **CNOT** is a two qubits operator. It has a control qubit and a target qubit. It works considering the value of the control qubit to apply the $\mathbf{X}$ operator on the target qubit. If the control qubit is set to $|1\rangle$ the $\mathbf{X}$ operator is applied to target qubit. The matrix representation for the **CNOT** in the computational basis is shown in Equation 3.

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{array}{l} \mathbf{CNOT}|00\rangle = |00\rangle \\ \mathbf{CNOT}|01\rangle = |01\rangle \\ \mathbf{CNOT}|10\rangle = |11\rangle \\ \mathbf{CNOT}|11\rangle = |10\rangle \end{array} \tag{3}$$

We can generalise and define a $(n+1)-$ary **CNOT** having n control qubits and requiring all the control qubits to be $|1\rangle$ for applying $\mathbf{X}$ on the target qubit.

Considering U as any quantum operator over n qubits, we can generalise the **CNOT** operator idea of control and target qubits further to a controlled-U gate, which has m controlled qubits and applies the U operator to the other n qubits if the controllers are all valued $|1\rangle$.

Any unitary quantum operator can be approximated by a sequence of tensor products of some of the 1-qubit gates seen above and the **CNOT** operator. We say that this set of gates is *universal* for quantum computation in the same vain that the NAND gate is universal for classical Boolean computation.

C. Quantum circuit

We can represent quantum operations by quantum circuits. This graphical representation considers the qubits as wires and quantum operators as boxes. The flow of the execution, as in the classical case, is from left to right.

Figure 1 has an example of a quantum circuit composed of a **CNOT**, where a filled circle depicts the control qubit, one $\mathbf{X}$ operator and one controlled-U operator where here $U = H \otimes H$.

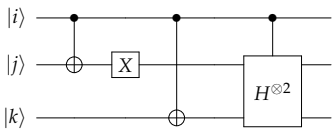


Fig. 1: An example of quantum circuit with two **CNOT** operators, one $\mathbf{X}$ operator and one controlled-U operator where U here is $H \otimes H$.

D. Quantum Fourier Transform

The quantum Fourier transform (QFT) algorithm takes into account the phase estimation of a given quantum state. The QFT is a tool widely used in quantum algorithms and essential step in the Shor algorithm, a

well-known algorithm in quantum computing that can find prime factors of an integer efficiently [22]. In this Section, we detail the QFT which is used in the recovery procedure of our proposed memory.

The QFT, inspired by the discrete Fourier transform, is the linear operator defined over an orthonormal basis $|0\rangle, \cdots, |N-1\rangle$ of a N-dimensional complex vector space, as:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle \tag{4}$$

In the particular case of n qubits system in the computational basis $|j\rangle = |j_1 j_2 ... j_n\rangle$, where $0 \leq j \leq 2^n - 1$, *i.e.* $N = 2^n$ in Equation 4 above, the QFT can be rewritten as the unitary operator defined as:

$$|j\rangle \xrightarrow{QFT} |j'\rangle = \frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle)...(|0\rangle + e^{2\pi i 0.j_1 j_2 ... j_n}|1\rangle)}{2^{n/2}} \tag{5}$$

The representation $0.j_l j_{l+1} ... jm$ means the binary fraction $\sum_{k=l}^{m} j_k 2^{-k}$. The QFT means that we can represent a given quantum state $|j\rangle$ in a new quantum state $|j'\rangle$ where its phase stores the information of the original quantum state $|j\rangle$. As the QFT is a unitary operator, its inverse operator can recover the information in the phase of a quantum state, transforming it into a respective quantum state. The definition of the QFT as defined in Equation 5 is called the product representation of the Fourier transform. From this representation it is immediate to generate the quantum circuit shown in Figure 2, where the operator R_x denotes the unitary transformation represented in Equation 6. In Figure 2, it was omitted for simplicity the last step procedure which is the swap operators to reverse the order of the qubits.

$$R_x = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^l} \end{bmatrix} \tag{6}$$

III. Quantum neuron model

The proposed quantum neuron circuit simulates any classical activation and propagation functions within a given precision. This circuit is shown in Figure 3. The quantum neuron is composed of two main components: a quantum operator $\mathbf{N}_\tau$ and the phase estimation by the $\mathbf{QFT^{-1}}$ operator. The main idea is to control the output of the neuron in the phase of quantum state depending on the input and weights. The $\mathbf{N}_\tau$ operator performs the quantum neuron functions. The $\mathbf{N}_\tau$ operator is defined as

$$\mathbf{N}_\tau(m,q,g,f) = \sum_{x_1=0}^{2^m-1} ... \sum_{x_n=0}^{2^m-1} \sum_{w_1=0}^{2^q-1} ... \sum_{w_n=0}^{2^q-1} Ph(\tau,g,f)|\mathbf{X},\mathbf{W}\rangle\langle\mathbf{X},\mathbf{W}| \tag{7}$$

where $Ph(\tau,g,f) = e^{\frac{2\pi i}{2^\tau} g(f(x_1,...,x_n,w_1,...,w_n))}$ is the phase shift element according the input and weights values and $\mathbf{X} = x_1, ..., x_n$ and $\mathbf{W} = w_1, ..., w_n$ are the possible values of input and weights respectively.

Let consider a neuron with n inputs $x_1, x_2, ..., x_n$ and weights $w_1, w_2, ..., w_n$. Each input x_i has a correspondent weight w_i. Each input and weight has p and q qubits respectively. The activation and propagation functions are respectively $f(x_1, x_2, ..., x_n, w_1, w_2, ...w_n)$ and $g(f(x_1, x_2, ..., x_n, w_1, w_2, ...w_n))$. The information generated by the quantum neuron is put on quantum phase and recovered by the quantum inverse Fourier transform operation.

The proposed quantum neuron operator $\mathbf{N}_\tau$ is parameterised by the τ value because during the recovery of the information processed by the neuron there exist some limitation of where to put this information. In other words, the τ value indicates the precision of the computation of the activation and propagation function. It must fit in a quantum state with a predefined number of qubits τ. The quantum neuron has also f and g functions as parameters, but we removed it of the subscription to not overload the notation.

The functions f and g transforms the input. They are connected such that output of the function f is the input of the function g. On its turns, the output of the function g will generate a phase of a quantum state with τ qubits to be estimated by the quantum inverse Fourier transform. The restriction on the functions f and g guarantees that their domain and codomain are compatible.

After the application of the operators $\mathbf{N}_\tau^{2^0}$, $\mathbf{N}_\tau^{2^1}$,...,$\mathbf{N}_\tau^{2^\tau-1}$ on the last two registers $|x_1, ..., x_n\rangle$ and $|w_1, ..., w_n\rangle$, the first register can be processed by the quantum inverse Fourier transform to capture the result of the quantum neuron output. This transformation is shown in Equation 8.

$$\begin{aligned} &\sum_{s=0}^{2^\tau-1} e^{2\pi i s.g(f(x_1,...x_n,w_1,..w_n))}|s\rangle|x_1,...x_n\rangle|w_1,...,w_n\rangle \\ &\xrightarrow{QFT^{-1}} |g(f(x_1,..x_n,w_1,...w_n)))\rangle|x_1,...x_n,w_1,...,w_n\rangle \end{aligned} \tag{8}$$

To deal with negative numbers, we consider that the first qubit in a quantum state represents the sign of the number. In this case, when a quantum state has m qubits, this means that we can represent the numbers from $-2^{m-1}-1$ to $2^{m-1}-1$. We use a superscription D in the number inside the *ket* and *bra* notation to indicate that we are considering the decimal representation with a sign. For example, we have the states $|011\rangle = |\mathbf{3}^D\rangle$ and $|111\rangle = |-\mathbf{3}^D\rangle$ with that representation. The decimal representation can also be used considering the representation $|j_1, j_2, ..., j_p\rangle = \sum_p j_n 2^{-p}$[23]. We present here three examples of activated and propagation functions to be implemented in the quantum neuron using this architecture. These examples can be extended to any function considering the structure of the domain and codomain of the neuron.

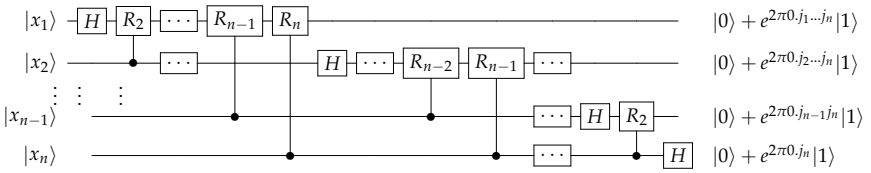


Fig. 2: Quantum circuit of the Quantum Fourier Transform

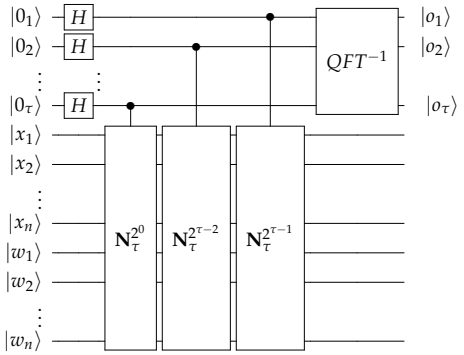


Fig. 3: Quantum neuron architecture. The operator $\mathbf{N}_\tau$ executes the nonlinear function of the neuron which is created as a function of the inputs $|x_1, ..x_n\rangle$ and weights $|w_1, ...w_n\rangle$ and puts the result on the first quantum register phase through successive τ Controlled-$\mathbf{N}_\tau$ operations. The QFT^{-1} operator recovers this phase information and transforms it in the quantum state $|o_1, ..., o_\tau\rangle$.

A. Inner product propagation function and linear activation function

We show here an example for a quantum neuron that has a linear activation and inner product between input and weight as propagation function. For this case, the domain of f is in $[-2^{p-1}-1, 2^{p-1}-1]^n \times [-2^{q-1}-1, 2^{q-1}-1]^n$ and its codomain is in $[-n(2^{p-1}-1)(2^{q-1}-1), n(2^{p-1}-1)(2^{q-1}-1)]$. The domain of g is in $[n(2^{p-1}-1)(2^{q-1}-1), n(2^{p-1}-1)(2^{q-1}-1)]$ and its codomain is in $[0, 2^\tau)$. Let consider $n = 3$ inputs/weights each one having $p = q = m = 4$ qubits. We have a function f that calculates the inner product between input and weights $\sum_n x_i w_i$ that generates the maximum value $n(2^{m-1}-1)(2^{m-1}-1) = 3(7)(7) = 147$ and minimum value -147. Considering that g is a linear function $g(x) = x$, then, the outputs to be processed by the neuron is 147 in maximum or -147 in minimum. This information is loaded in a state with at least 9 qubits (1 for sign and 8 to the number). Then, τ value chosen is $\tau = 9$. The recovery procedure using quantum Fourier transform will recovery the state between the state $|010010011\rangle = |\mathbf{147}^D\rangle$ to the state $|110010011\rangle = |-\mathbf{147}^D\rangle$.

B. Step activation function

Let consider that the propagation function is the inner product between inputs and its corresponding weights. For $n = 3$ inputs and $m = 4$, the inner product is in maximum value 147 and minimum value -147, as we mentioned in the previous propagation type. The step propagation function is a function that for values above a threshold (normally 0) the output is 1, and 0 otherwise. We can set the propagation function $g(x) = 1$ for $x \geq 0$ and $g(x) = 0$ otherwise.

C. Sigmoid activation function

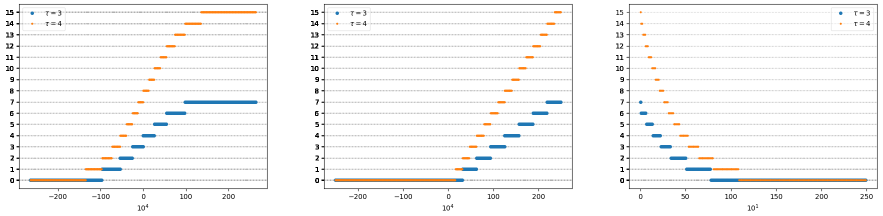
The sigmoid function is $\frac{1}{1+e^{-x}}$. Including this function to operate in the neuron, its maximum value is $2^{\tau-1}-1$, because it is the maximum positive value calculated in a quantum state with τ qubit. Then the sigmoid activation function can be a discretisation of the function $g(x) = \frac{2^{\tau-1}-1}{1+e^{-x}}$.

D. Radial basis activation and propagation function

Radial basis function (RBF) networks have neuron with functions which consider distances of centroids. Having one hidden layer, RBF networks are capable of universal approximation [24]. The radial basis functions output the distance of the input from a central vector. In other words, one calculates $\sqrt{\sum_i (x_i - c_i)^2}$ for each input x_i and its correspond ith central value c_i. We can consider the neuron weights as the central vector. Then the propagation function is $f(x_1, ..., x_n, w_1, ..., w_n) = \sqrt{\sum_i (x_i - w_i)^2}$. The radial basis activation function normally used is the exponential with some variance σ, $g(x) = (2^{\tau-1}-1)e^{\frac{x^2}{\sigma^2}}$ for a given τ value.

E. ReLU activation function

Rectified linear (ReLU) function has been used in the training of deeper networks [25], [26] as activation function of neurons combined with the inner product between input and weights as propagation function. The ReLu function is defined as $g(x) = max(0, x)$. As the neuron output has precision values, the ReLu



(a) (b) (c)

Fig. 4: Graphs of (a) sigmoid, (b) ReLu and (c) exponential propagation functions for $\tau = 3$ and $\tau = 4$.

function can be defined in the quantum neuron as $g(x) = max(0, (2^\tau - 1)x/(n(2^{p-1} - 1)(2^{q-1} - 1)))$ for the τ qubits of output precision, p and q qubits of inputs and weights precision respectively and n inputs and weights.

F. Examples

The well known XOR problem [1] - to set an ANN to compute the exclusive OR boolean function - is hard for ANNs. Networks with one layer of Perceptrons cannot compute the XOR function or any other function which is not linearly separable [27]. However, a general single artificial neuron can compute XOR depending on the activation function.

Let us consider the input register $|\mathbf{x}\rangle$ and weights register $|\mathbf{w}\rangle$ with 2 qubits. As the possible values for the output is either 0 or 1, it is enough τ be set to 1, since there are $2^\tau = 2$ different expected states. We configure the weights to be $|11\rangle$ and then the neuron architecture is defined to linear propagation function, and the activation function as $g_{ex1}(x) = 1$ when $x \in \{3, 6\}$ and 0 otherwise. When the input is either 00 or 11 the output of the f_{ex1} function is 0 and 2 respectively, and the output of the neuron is 0 in both case. When the input is either 01 or 10 the f_{ex1} output is 1 in both case and the neuron output is then 1. The neuron operator is shown in Equation 9 represents this operation.

$$\mathbf{N}_1 = \sum_{\mathbf{x}=01,10} e^{\frac{2\pi i}{2}} |\mathbf{x}, 11\rangle\langle\mathbf{x}, 11| + \sum_{\mathbf{x}\neq 01,10} |\mathbf{x}, 11\rangle\langle\mathbf{x}, 11| + \sum_{\mathbf{x}} \sum_{\mathbf{w}\neq 11} |\mathbf{x}, \mathbf{w}\rangle\langle\mathbf{x}, \mathbf{w}| \tag{9}$$

Consider now a neuron with ten inputs and weights, each one having ten qubits where the first qubit represents the negative sign. For the inner product propagation function, the possible minimum and maximum values are $-10 * 2^9 = -2621440$ and $10 * 2^9 = 2621440$ respectively. With the sigmoid and ReLu activation functions g we have Figures 4a and 4b respectively. The neuron operator is $\mathbf{N}_\tau(m = 10, q = 10, g_{ex1}, f_{ex1})$.

Another example is the quantum neuron with RBF as propagation function. It represents in its weights the values of a centroid. We can see the radial basis functions as a propagation function f_{ex2} and the exponential with variance equal to one to be the activation function g_{ex2}. The centroid components w_1, w_2, w_3, w_4, w_5 are represented in the quantum states $|w_1\rangle, |w_2\rangle, |w_3\rangle, |w_4\rangle, |w_5\rangle$. Assume that each input and weight has ten qubits. The neuron circuit for this example is $\mathbf{N}_\tau(m = 10, q = 10, g_{ex2}, f_{ex2})$.

The maximum value of the distance calculated by the neuron is $\sqrt{5 * (1023^2)} \approx 2287.50$ and the minimum distance is 0 for a given input set. The graphics of the function g is shown in Figure 4c for $\tau = 3$ and 4.

The computational complexity of the quantum generalised neuron is $O(n)$, where n is the input size. The quantum inverse Fourier transform requires $\frac{\tau(\tau+1)}{2} + 3\frac{\tau}{2}$ gates [2] and the τ operations are done before the calculus of the phase estimation. The process of training the neuron weights to fit for some training dataset can use the quantum parallelism. This can be done using existing quantum training algorithms [28], [29], [8] or quantum-based algorithms [30]. The process to search the best architecture for the quantum neural network is shown in [31].

IV. Conclusion

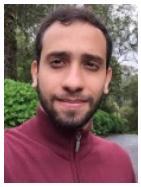
In this paper, we presented a novel generalised architecture for a weighted quantum neuron parametrised by propagation and nonlinear activation functions. The input and weights size as well the size of output precision are also parameters of the neuron. The neuron can process parallel information and calculate in a single step the output of the neuron for superposed inputs and weights. The quantum inverse Fourier transform operator is used to recover the result encoded in the output of the neuron. The new quantum neuron is a model which can be used as a quantum machine learning unit to load and recover information.

Acknowledgment

This work was supported by the Serrapilheira Institute (grant number Serra-1709-22626) and CNPq, Brazilian research agency, (Edital Universal, grant number 421849/2016-9 and 442668/2014-7).

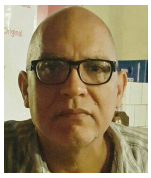
REFERENCES

[1] S. Haykin, *Neural Networks*. Prentice Hall, 1999.
[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195 EP –, Sep 2017.
[4] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.
[5] ——, "The quest for a quantum neural network," *Quantum Information Processing*, vol. 13, no. 11, pp. 2567–2586, Nov. 2014.
[6] S. Gupta and R. Zia, "Quantum neural networks," *Journal of Computer and System Sciences*, vol. 63, no. 3, pp. 355 – 383, 2001.
[7] W. R. de Oliveira, A. J. da Silva, and T. B. Ludermir, "Vector space weightless neural networks," in *ESANN 2014*, no. April, 2014, pp. 535–540.
[8] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir, "Classical and superposed learning for quantum weightless neural networks," *Neurocomputing*, vol. 75, no. 1, pp. 52–60, 2012.
[9] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *International Journal of Circuit Theory and Applications*, vol. 39, no. 1, pp. 61–77, 2011.
[10] R. Zhou and Q. Ding, "Quantum m-p neural network," *International Journal of Theoretical Physics*, vol. 46, no. 12, pp. 3209–3215, 2007.
[11] M. V. Altaisky, "Quantum neural network," Joint Institute for Nuclear Research, Russia, Technical report, 2001.
[12] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir, "Comments on "quantum m-p neural network"," *International Journal of Theoretical Physics*, vol. 54, no. 6, pp. 1878–1881, 2015.
[13] E. Behrman, L. Nash, J. Steck, V. Chandrashekar, and S. Skinner, "Simulations of quantum neural networks," *Information Sciences*, vol. 128, no. 3–4, pp. 257 – 269, 2000.
[14] M. Schuld, I. Sinayskiy, and F. Petruccione, "Simulating a perceptron on a quantum computer," *Physics Letters A*, vol. 379, no. 7, pp. 660 – 663, 2015.
[15] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, "Quantum neuron: an elementary building block for machine learning on quantum computers," *arXiv preprint arXiv:1711.11240*, 2017.
[16] N. Wiebe and V. Kliuchnikov, "Floating point representations in quantum circuit synthesis," *New Journal of Physics*, vol. 15, no. 9, p. 093041, 2013.
[17] W. Hu, "Towards a real quantum neuron," *Natural Science*, vol. 10, no. 03, p. 99, 2018.
[18] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *arXiv preprint arXiv:1806.06871*, 2018.
[19] V. Gandhi, G. Prasad, D. Coyle, L. Behera, and T. M. McGinnity, "Quantum neural network-based eeg filtering for a brain–computer interface," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 278–288, 2014.
[20] T. Menneer and A. Narayanan, "Quantum-inspired neural networks," *Tech. Rep. R329*, 1995.
[21] S. Jeswal and S. Chakraverty, "Recent developments and applications in quantum neural network: A review," *Archives of Computational Methods in Engineering*, pp. 1–15, 2018.
[22] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.
[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
[24] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, June 1991.
[25] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 855–863.
[26] C. Ding, Y. Li, L. Zhang, J. Zhang, L. Yang, W. Wei, Y. Xia, and Y. Zhang, "Fast-convergent fully connected deep learning model using constrained nodes input," *Neural Processing Letters*, Jun 2018.
[27] M. Minsky and S. A. Papert, *Perceptrons: an introduction to computational geometry*. MIT press, 2017.
[28] B. Ricks and D. Ventura, "Training a quantum neural network," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, ser. NIPS'03. Cambridge, MA, USA: MIT Press, 2003, pp. 1019–1026.
[29] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *International Journal of Circuit Theory and Applications*, vol. 39, no. 1, pp. 61–77, 2011.
[30] T.-C. Lu, G.-R. Yu, and J.-C. Juang, "Quantum-based algorithm for optimizing artificial neural networks," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 8, pp. 1266–1278, 2013.
[31] A. J. da Silva and R. L. F. de Oliveira, "Neural networks architecture evaluation in a quantum computer," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, Oct 2017, pp. 163–168.



Fernando M de Paula Neto received the Bachelor degree (cum laude) in Computer Engineering in 2014 and the Master degree in Computer Science in 2016 at Universidade Federal de Pernambuco (Brazil). He is currently working toward the Ph.D. degree in Computer Science at the Universidade Federal de Pernambuco, Brazil. His current research interests include Quantum Computing, Quantum Machine Learning, Dynamical Systems and Chaos, Neural Networks and Hybrid Intelligent Systems.

Teresa Bernarda Ludermir received the Ph.D. degree in Artificial Neural Networks in 1990 from Imperial College, University of London, UK. From 1991 to 1992, she was a lecturer at Kings College London. She joined the Center of Informatics at Federal University of Pernambuco, Brazil, in September 1992, where she is currently a Professor and head of the Computational Intelligence Group. She has published over 300 articles in scientific journals and conferences, three books in Neural Networks and organized two of the Brazilian Symposium on Neural Networks. Her research interests include weightless Neural Networks, hybrid neural systems and applications of Neural Networks.



Wilson R de Oliveira received his Ph.D. (2004) in Computer Science at Universidade Federal de Pernambuco (Brazil). He was on sabbatical leave at the School of Computer Science, University of Birmingham, UK (2008).He has been working recently on Quantum Weightless Neural Networks, Categorical Quantum Mechanics and Discrete Manifolds. He joined the "Departmento de Estatística e Informàtica" at the "Universidade Federal Rural de Pernambuco" in 2000 where he is now an Associate Professor.

Adenilton J da Silva received the Licenciate degree (cum laude) in mathematics from Universidade Federal Rural de Pernambuco (UFRPE), Brazil and received the Ph.D. degree in Computer Science from Universidade Federal de Pernambuco, Brazil. He joined the Departamento de Estatística e Informàtica at UFRPE in 2014 where is now Assistant Professor. His current research interests include quantum computation, artificial neural networks, machine learning and hybrid neural systems.

APÊNDICE C – QUANTUM NEURONS WITH INTERNAL MEMORY

Quantum neurons with internal memory

Fernando M de Paula Neto[a,*], Wilson R de Oliveira[b], Teresa B. Ludermir[a]

[a] *Universidade Federal de Pernambuco, Cidade Universitária, Recife, Pernambuco, Brazil*
[b] *Universidade Federal Rural de Pernambuco, Dois Irmãos, Recife, Pernambuco, Brazil*

Abstract

We investigate general quantum models of neurons that carry internal information and update themselves when they are excited with an input. The models consider the neuron interaction with the environment (nonlinear operators) and other neurons. The internal information feature gives the neuron the possibility to process temporal information. Numerical experiments show the parametric sensibility of a quantum operator coupled with different environment linear and nonlinear operators, showing chaotic behaviour and bifurcation in its dynamics.

Keywords: Quantum neuron, Quantum nonunitary operators, Quantum neuron with Internal memory, Quantum memory, Environment interaction, Open quantum computing, Dynamical systems

1. Introduction

The recurrent neural networks have been used to map temporal information and good results are encountered in many applications as time series forecasting tasks [1] or handwritten letters recognition [2]. Some evidences were encountered that biological neurons have dynamical properties during their execution as chaos and dynamics internal state [3]. Experiments with squid giant axons and numerically with the Hodgkin-Huxley equations show that neurons responses are not always periodic and that the apparently nonperiodic responses can be understood as deterministic chaos [4]. Through this evidences some artificial neurons are modeled to have those properties. It is also aimed to understand the parametric behaviour of this dynamics in neural networks models [5].

Quantum computing field investigates information processing considering the quantum mechanics theory. There are quantum models for simulating algorithms in machine learning using intrinsically quantum features [6]. Quantum neurons were proposed to model the behaviour of classical neurons considering specific nonlinear functions [7, 8, 9, 10, 11, 12] and general nonlinear function parametrized neuron [13]. In [14], a Perceptron with internal memory was proposed. In this paper, we extend the concept of internal memory for a quantum neuron with any nonlinear function and with interaction with the environment. No one existing neuron network proposed before leads with general neuron model with internal memory updating and its interaction with nonlinear operators.

This paper is organized as follows. Section 2 describes the quantum computation theory, including the nonlinear operators description. Section 3 talks about the existing quantum neurons models. The proposed model is detailed in Section 4. The conclusions are done in Section 5.

*Corresponding author
Email address: fmpn2@cin.ufpe.br (Fernando M de Paula Neto)

2. Quantum computation

2.1. Quantum bits

The unit of information in the quantum computation is called a *quantum bit* (qubit). The qubit is a two-dimensional vector in the complex vector space $\mathbb{C}^2$. It can be in superposition of states, *ie.* in the position 0 or in the position 1 in same time, if we consider the canonical basis as 0 and 1. Any qubit $|\psi\rangle$ can be written as a linear combination of vectors (or states) of $\mathbb{C}^2$ canonical (or computational) basis $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$ as viewed in Equation 1,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are complex number and $|\alpha|^2 + |\beta|^2 = 1$. This notation also means that the qubit has $|\alpha|^2$ to be measured as 0 and $|\beta|^2$ to be measured as 1.

The qubits are represented mathematically united by the tensor operator, $\otimes$. The tensor operator is used to represent quantum systems with two or more qubits $|\mathbf{g}\rangle = |ij\rangle = |i\rangle \otimes |j\rangle$. Here we will use the bold font for the representation of quantum states with more one qubit. For two qubits $|i\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|j\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, the tensor operator generates the state $|ij\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle$. For a general two states $|\mathbf{p}\rangle$ and $|\mathbf{q}\rangle$ with n and m states respectivally, the state $|pq\rangle$ can be calculated by the operation described in Equation 2.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ ... \\ \alpha_{2^n} \end{bmatrix} \otimes \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_{2^m} \end{bmatrix} = \begin{bmatrix} \alpha_1 \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_{2^m} \end{bmatrix} \\ \alpha_2 \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_{2^m} \end{bmatrix} \\ ... \\ \alpha_{2^n} \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_{2^m} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_2 \\ ... \\ \alpha_1\beta_{2^m} \\ \alpha_2\beta_1 \\ \alpha_2\beta_2 \\ ... \\ \alpha_2\beta_{2^m} \\ ... \\ \alpha_{2^n}\beta_{2^m} \end{bmatrix} \tag{2}$$

We can represent the quantum states using integer numbers rather than string bits inside the $|.\rangle$ notation. For a given quantum state with n states $|\boldsymbol{\psi}\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdot + \alpha_n|n\rangle$ the measurement of the $|x\rangle$ state may occur with $|\langle\boldsymbol{\psi}||x\rangle|^2$ of probability where the $\langle.|$ represents the complex conjugate of the vector $|.\rangle$.

Let Q and R be two vector spaces the tensor product of Q and R, denoted by $Q \otimes R$, is the vector space generated by the tensor product of all vectors $|a\rangle \otimes |b\rangle$, with $|a\rangle \in A$ and $|b\rangle \in B$. Some states $|\psi\rangle \in Q \otimes R$ cannot be written as a product of states of its component systems Q and R. States with this property are called *entangled*

states, for instance two entangled qubits are the Bell states described in Equation (3).

$$\begin{aligned} \left|\Phi^{+}\right\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ \left|\Phi^{-}\right\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\ \left|\Psi^{+}\right\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ \left|\Psi^{-}\right\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \tag{3}$$

2.2. Quantum operators

The quantum states are modified by quantum operators which change the amplitude values of the qubits. A quantum operator **U** over n qubits is a unitary complex matrix of order $2^n \times 2^n$. For example, some operators over 1 qubit are: Identity **I**, NOT **X** and Hadamard **H**, described below in Equation (4) and Equation (5) in matrix form and operator form. The combination of these unitary operators forms a quantum circuit.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{array}{l} \mathbf{I}|0\rangle = |0\rangle \\ \mathbf{I}|1\rangle = |1\rangle \end{array} \quad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{array}{l} \mathbf{X}|0\rangle = |1\rangle \\ \mathbf{X}|1\rangle = |0\rangle \end{array} \tag{4}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{array}{l} \mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle) \end{array} \tag{5}$$

The Identity operator **I** generates the output exactly as the input; **X** operator works as the classic NOT in the computational basis; Hadamard **H** generates a superposition of states when applied in a computational basis.

In the same way we can combine quantum states, quantum operators can also be combined using tensor product. For two (n_0, m_0)-dimensional matrix U and (n_1, m_1)-dimensional matrix V, their composition, $U \otimes V$, products a third $(n_0 n_1, m_0 m_1)$-dimensional matrix. We denote as $\mathbf{A}^{\otimes s}$ the s-fold application of **A**.

The **CNOT** is a two qubits operator. It has a control qubit and a target qubit. It works considering the value of the control qubit to apply the **X** operator on the target qubit. If the control qubit is set to 1 the **X** operator is applied to target qubit. The matrix representation for in the computational basis is shown in Equation 6.

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{array}{l} \mathbf{CNOT}|00\rangle = |00\rangle \\ \mathbf{CNOT}|01\rangle = |01\rangle \\ \mathbf{CNOT}|10\rangle = |11\rangle \\ \mathbf{CNOT}|11\rangle = |10\rangle \end{array} \tag{6}$$

We can generalise and define a $(n+1)-$ary **CNOT** having n control qubits and requring all the control qubits to be $|1\rangle$ for applying **X**.

2.3. Quantum circuit

We can represent quantum operations by quantum circuits. This graphical representation considers the qubits as wires and quantum operators as boxes. The flow of the execution, as in the classical case, is from left to right.

Figure 1 has an example of a quantum circuit composed of a **CNOT**, where the control qubit is depicted by a filled circle, and a **X** operator.

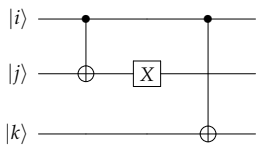


Figure 1: An example of quantum circuit with two **CNOT** operators and a **X** operator.

2.4. Quantum Parallelism

The quantum algorithms can calculate in superposition of input basis states the value of a function. This means that a function can be evaluated by all its possible inputs in a single time. This characteristic feature of quantum computation is called quantum parallelism. Let consider the U_f operator which maps $|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$, where $\oplus$ indicates modulo-2 addition. If we pass as input the superposition of state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ the result of this function is $\frac{|0,f(0)\rangle+|1,f(1)\rangle}{\sqrt{2}}$. This operation is shown in Figure 2.

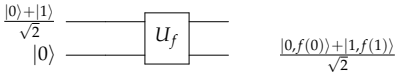


Figure 2: An example of quantum parallelism in a quantum circuit.

2.5. Nonunitary quantum computation

To measure a quantum state is an irreversible operation that partial or totally loses the information about a superposition of states. For example, in a qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, a measurement collapses (projects) the state either to $|0\rangle$ state with $|\alpha|^2$ of probability or to $|0\rangle$ with $|\beta|^2$ probability. For a composed and superposed quantum state $|\mathbf{g}\rangle$ a probability to see a state $|\mathbf{i}\rangle$ is $|\langle \mathbf{g}|\mathbf{i}\rangle|^2$. In Figure 3 an example of the measurement operator in a quantum circuit is shown.

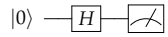


Figure 3: Measurement results 0 or 1 with equal probability $\frac{1}{2}$.

Other non-unitary operators have been proposed and their effects in quantum computation analysed. Non-unitary and non-linear operators [15][16][17][18] combined with quantum gates solve analytically NP-complete problems in polynomial time [19][20][21]. Quantum neural networks can be trained by non-linear operators [12, 22]. Nonlinear effects in quantum dynamics as chaos [23] can be observed when non-unitary operators are used. Although there are controversies about the physical realisation of non-unitary operators, in [17, 24] a quantum unitary operator is embedded in a unitary operator and can be applied probabilistically.

In [25], it was shown that any nonunitary quantum circuit can be represented by a set of unitary and measurement operators. It is possible because the singular value decomposition allows the representation of a $2^n \times 2^n$ nonunitary matrix operator N in three components: 2 unitary matrices U and V, and one diagonal matrix with 2^n

components, D. Then a matrix N can be equal to the product of matrices UDV. The unitary matrices can be built using CNOT gates and one-qubit unitary gates, since this set of operators is universal for the unitary quantum circuit [26]. The diagonal matrix $D(d_1, d_2, ..., d_{2^n})$ can be factorized into 2^n matrices in the form of $D(1, 1, ..., d_i)$. One can represent the matrix $D(1, 1, ..., d_i)$ by the Controlled-$N_1(a)$ gate with $n-1$ control qubits where $N_1(a)$ is a one-qubit unitary gate given by

$$N_1(a) = \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix}, 0 \leq a < 1 \, . \tag{7}$$

In [25] it is shown that the Controlled-$N_1(a)$ can still be decomposed into one $N_1(0)$ gate, which is a one-qubit projective measurement $|0\rangle\langle 0|$ and one Controlled-$U_1(a)$ gate, where

$$U_1(a) = \begin{pmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{pmatrix} . \tag{8}$$

Some nonunitary operators have been used to solve problems in quantum computing. For example, a nonlinear operator which appears during time evolution of quantum states is capable to solve NP-Complete problems in polynomial times [27]; measurement operators are used to solve NP-Complete problems with probabilistic or exactly operation whether they are implementable [23, 18]. The open quantum systems field leads with the possibility to interact quantum unitary operators with the nonlinearity presents in almost all systems in the Universe. We select 4 nonunitary operators, that have been used in quantum algorithms before. Here these operators are detailed and their function described.

2.5.1. Abrams and Lloyd operator

Abrams and Lloyd [27] propose to use a nonlinear operator considering that quantum states can exhibit small nonlinearities involving during its time evolution. This operator has been used to solve NP complete problems in polynomial time and for content recovery in associative memory [28, 29].

The operation of this nonlinear operator is like an AND gate. Considering the first qubit as index qubit and the second one as flag qubit, the operator put $|1\rangle$ in the flag qubit if some of the flag qubit is $|1\rangle$ and keeps $|0\rangle$ otherwise. The index qubit is not altered. In [27] it is shown the transformation of three possible quantum states encountered solving a NP complete problem:

$$\begin{aligned} |00\rangle + |11\rangle &\longrightarrow |01\rangle + |11\rangle \\ |01\rangle + |10\rangle &\longrightarrow |01\rangle + |11\rangle \\ |00\rangle + |10\rangle &\longrightarrow |00\rangle + |10\rangle \end{aligned} \tag{9}$$

This nonlinear transformation can be built using unitary operations and single qubit nonlinear operators. In [27], a possible technique to generate this transformation is the action on two qubits unitary operator:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} . \tag{10}$$

Considering the three possible quantum states described in Equation 10, this unitary transformation transforms the states above as

$$\begin{aligned} |00\rangle + |11\rangle &\longrightarrow |00\rangle \\ |01\rangle + |10\rangle &\longrightarrow |01\rangle \\ |00\rangle + |10\rangle &\longrightarrow \frac{1}{2}[|00\rangle + |01\rangle - |10\rangle + |11\rangle] \end{aligned} \tag{11}$$

After that, the authors propose to use a simple one qubit nonlinear gate $\hat{n}_-$ that maps both $|0\rangle$ and $|1\rangle$ to the state $|0\rangle$:

$$\begin{aligned} |00\rangle + |11\rangle &\longrightarrow |00\rangle \\ |01\rangle + |10\rangle &\longrightarrow |00\rangle \\ |00\rangle + |10\rangle &\longrightarrow |A\rangle \end{aligned} \tag{12}$$

The third final state is unknown because it depends on the implementation of $\hat{n}_-$ operator over the third case. The authors point out that this omsision allows for flexibility in the choice of $\hat{n}_-$. The expected expression of $|A\rangle$ is $|0\rangle(x|0\rangle + y|1\rangle)$.

A second nonlinear gate $\hat{n}_+$ maps the state $x|0\rangle + y|1\rangle$ to the state $|1\rangle$:

$$\begin{aligned} |00\rangle + |11\rangle &\longrightarrow |00\rangle \\ |01\rangle + |10\rangle &\longrightarrow |00\rangle \\ |00\rangle + |10\rangle &\longrightarrow |01\rangle \end{aligned} \tag{13}$$

To complete the nonlinear transformation, a NOT gate is applied on the second qubit and a Hadamard gate on the second one, resulting in the operation described in Equation 9.

2.5.2. Leporati and Felloni operator

In [18], a non-unitary operator is proposed to solve a NP complete problem in polynomial time. This operator exists considering two conditions of implementation: if it is possible to execute $2^n|1\rangle\langle 1|$ to a qubit in a quantum circuit and if an external observer is able to distinguish between a null and a non-null vector. The exponential number, 2^n, in the circuit is used to amplify the amplitude of the qubit $|1\rangle$. This amplification step is inspired from a chaotic approach [30] in which iteratively allows small values of amplitudes to be amplified through a chaotic map.

This operator is described as

$$O = 2^n \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{14}$$

and has two possible outcomes (a) the *null* vector $\mathbf{0}$, if a quantum state is $|0\rangle$, because $O|0\rangle = 2^n|1\rangle\langle 1||0\rangle = \mathbf{0}$ or (b) a non-null vector if the quantum state is different of $|0\rangle$, since $O(\alpha_0|0\rangle + \alpha_1|1\rangle) = \alpha_0 2^n|1\rangle\langle 1||0\rangle + \alpha_1 2^n|1\rangle\langle 1||1\rangle = \mathbf{0} + \alpha_1 2^n|1\rangle = \alpha_1 2^n|1\rangle$.

Using this operator, previous algorithms were been proposed to solve NP complete problems [21].

2.5.3. Bechmann-Pasquinucci, Huttner and Gisin operator

Applied to distinguish optimizally between two non-orthogonal states, the nonlinear quantum state transformation presented in [31] proposes to use a CNOT operator and a non-unitary operator to square the amplitudes of a given quantum state. The nonunitary operator used in this operator is of the same class of Leporati and Felloni operator. The circuit of the nonlinear operator is shown in Figure 4. The procedure gives in account that it is possible to feed the circuit with the same quantum state in the first and the second wires. After the CNOT operation, a measurement of the second particle is done in which it keeps the quantum state if the result is $|0\rangle$.

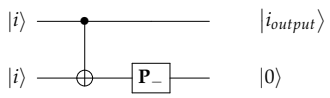


Figure 4: Quantum circuit of the Bechmann-Pasquinucci *et. al* nonlinear operator, where $\mathbf{P}_- = |0\rangle\langle 0|$.

Then this operation transforms the amplitudes of a qubit in squared amplitudes:

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \alpha^2|0\rangle + \beta^2|1\rangle \tag{15}$$

This operator has been used to encounter chaos in quantum state dynamics [32, 23] since nonlinear transformation is a condition to observe chaotic behaviour.

2.5.4. Quadratic measurement operator

During the iteration of quantum weightless neurons, it was shown in [33] that the quantum output of the neuron could be entangled. Then it was proposed to use several times the measurement operator to build the amplitudes of a quantum output state before it be fedback. These reconstructed amplitudes are always real values because they appear after a successive measurements.

This procedure is possible because the states have probability of occurrence in function of the amplitudes. For a given quantum state $\alpha|0\rangle + \beta|1\rangle$ where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$, one sees the state $|0\rangle$ with $|\alpha|^2$ of chance and the state $|1\rangle$ with $|\beta|^2$. So, the quantum state is measured and the process repeated sometimes. With this routine, it is possible to mount the observed probability amplitudes representing the unknown quantum state.

Analytically this is can be viewed by an example. Given the states $|i\rangle$ and $|j\rangle$

$$|i\rangle = \begin{pmatrix} a \\ b \end{pmatrix}, |j\rangle = \begin{pmatrix} c \\ d \end{pmatrix} \tag{16}$$

After the application of any unitary operator $U|\mathbf{ij}\rangle$, the resulting quantum state can be entangled or not, and it can be represented as $|ij\rangle' = e|00\rangle + f|01\rangle + g|10\rangle + h|11\rangle$.

This operator rebuilds the output qubits considering the probability to check either zero or one in the first qubit, and the same operation to the second qubit. Then the amplitudes are rebuilt as

$$\alpha_1^{t+1} = \sqrt{|e|^2 + |f|^2}, \quad \beta_1^{t+1} = \sqrt{|g|^2 + |h|^2} \tag{17}$$

$$\alpha_2^{t+1} = \sqrt{|e|^2 + |g|^2}, \quad \beta_2^{t+1} = \sqrt{|f|^2 + |h|^2}. \tag{18}$$

In this way, the qubits in the next iteration are built using the amplitude values encountered by the repetition of the measurement procedure. Then, they are generate as

$$|i\rangle_{t+1} = \alpha_1^{t+1}|0\rangle + \beta_1^{t+1}|1\rangle \tag{19}$$

$$|j\rangle_{t+1} = \alpha_2^{t+1}|0\rangle + \beta_2^{t+1}|1\rangle. \tag{20}$$

In [23], it was shown that this extraction iterated procedure generates chaos and bifurcation in the dynamics of a quantum neuron model for a given set of input states.

3. Quantum neurons models

There are some quantum neurons models inspired in the operation of the biological neuron. The main challenge in quantum computing is that the quantum operations are unitary and the neuron carries a nonlinear operator in this working. A description of a quantum neural network model is given in [9]. In this model, the neural network saves its information in a quantum operator. For a given quantum system with n inputs $|x_1\rangle \dots |x_n\rangle$, the output of this neuron is $|y\rangle = \hat{F}\sum_{j=1}^{n} \hat{w}_j|x_j\rangle$ where $\hat{w}$ is a 2x2 matrix operator which saves the neuron information. The modification of this $\hat{w}$ quantum operator is done by an iterative learning process described in Equation 21 where $|d\rangle$ is the desired state.

$$\hat{w}_j(t+1) = \hat{w}_j(t) + \eta(|d\rangle - |y(t)\rangle)\langle x_j| \tag{21}$$

It was demonstrated that the learning rule described in [9] does not preserve the unitary of the operators [34]. A quantum neuron whose weights are into a quantum operator was also proposed in [35]. In this model, the learning algorithm adjusts the weights of the neuron according the expected output. This operation is described in Equation 22.

$$w_{ij}^{t+1} = w_{ij}^{t} + \eta(|O\rangle_i - |\psi\rangle_i)|\phi\rangle_j \tag{22}$$

The w_{ij} weights are the matrix entries indexed by row i and column j; η is the learning rate, $|\phi\rangle$ and $|O\rangle$ are respectively examples of input and expected output in the training set, W is the weight matrix of the neuron and $|\psi\rangle$ the application of the input in the weight matrix $|\psi\rangle = W^t|\phi\rangle$. It was shown that this neuron can be efficiently simulated in a classical neural networks [36].

In [37], it was proposed to use a nonlinear operator to find the best parameters of a neural network in a superposition way. The quantum architecture allows to use the parallelism to evaluate all the possible weights of the networks and the nonlinear proposed in [27] makes an exhaustive search of the optimal parameters. In [38] and [39], the quantum RAM based neuron was defined as the quantisation of the weightless neural networks proposed in [40]. The RAM node stores in its memory one bit addressed by an input bit string. The qRAM represents that bit storage by the gate A, that is a **CNOT** gate. A quantum neuron viewed over a perspective of the time evolution of

a single quantum object is demonstrated in [41]. The neuron as a memory which recoveries the information based on Grover's search algorithm is proposed in [42].

Another model of the quantum perceptron is proposed in [7]. This model simulates only a classical perceptron which has the step function as activation function. It uses the inverse of the quantum Fourier transform algorithm to calculate the inner product between the input and the weights of the neuron. The step function is simulated when the first qubit of the inner product result is measured. If this qubit is 1, it is expected the inner product is more than $1/2$. The probability of success depends on the precision of qubits we use to represent the inner product. In Figure 5, it is shown the quantum circuit which calculates the inner product with the weights w and the input $|x\rangle$. In this model, the weights are fixed in quantum operators. The algorithm starts with some τ zeroes, in which the τ is the precision used, and the input $|\psi_0\rangle = |x_1, x_2, ..., x_n\rangle$. The operator $U_k(w_k)$, described in Equation 23, where $\Delta_\phi = \frac{1}{2n}$, applies the phase change by kth input register qubit.

$$U_k(w_k) = \begin{pmatrix} e^{-2\pi i w_k \Delta_\phi} & 0 \\ 0 & e^{2\pi i w_k \Delta_\phi} \end{pmatrix} \tag{23}$$

Then the inverse quantum Fourier transform calculates the phase included in the input state through the $U(\hat{w}) = U_n(w_n)... \otimes ...U_2(w_2) \otimes U_1(w_1)$ operator. $|J\rangle = |J_1, ..., J_\tau\rangle$ is the binary representation of the integer j and $\phi = \frac{j}{2^\tau}$. For the state be in the way that we can calculate the estimation of the inner product of the input with the weight vector, the algorithm applies sometimes the controlled-$U(\hat{w})^{2^j}$ operation (called *modular exponentiation*) to get the result quantum state in the shape where the inverse quantum Fourier transform can calculate the phase estimation correctly. The modular exponentiation can be done using the trick of uncomputation and details of implementation are encountered in [26]. In Equation 24 the operation of quantum inverse Fourier is shown where the phase ϕ is estimated as $\hat{\phi}$.

$$\frac{1}{\sqrt{2^\tau}} \sum_{j=0}^{2^\tau - 1} \exp^{2\pi i j \phi} |J\rangle |\psi_0\rangle \xrightarrow{QFT^{-1}} |\hat{\phi}\rangle |\psi_0\rangle \tag{24}$$

Schuld and collaborators [7] also introduce a slight variation of this quantum perceptron including the weights as quantum registers. This allows to put the weights also as an input of the circuit. The initial state considers the input and the weights of the neuron

$$\left| x_1, ..., x_n; W_1^{(1)}, ..., W_1^{(\delta)}, ..., W_n^{(1)}, ..., W_n^{(\delta)} \right\rangle = |\mathbf{x}; \mathbf{w}\rangle \tag{25}$$

$W_k^{(m)}$ is the mth digit of the binary fraction representation that express w_k as $w_k = W_k^1 \frac{1}{2} + ... + W_k^{(\delta)} \frac{1}{2^\delta}$ with a precision δ. For this modification, it is introduced the controlled two-qbuit operator

$$U_{w_k^{(m)}, x_k} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-2\pi i \Delta_\phi \frac{1}{2^m}} & 0 \\ 0 & 0 & 0 & e^{2\pi i \Delta_\phi \frac{1}{2^m}} \end{pmatrix} \tag{26}$$

The mth bit $W_k^{(m)}$ of the binary representation of w_k controls the operation of shifting the phase by $\Delta\phi \frac{1}{2^m}$ (for $x_k = 0$) or $\Delta\phi \frac{1}{2^m}$ (for $x_k = 1$), using $\Delta\phi$ from above. For more details, see [7].

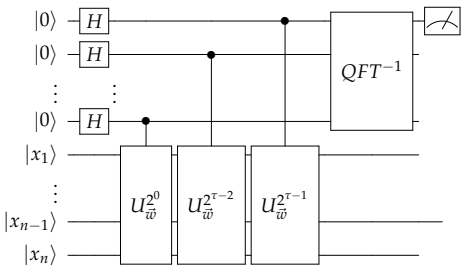
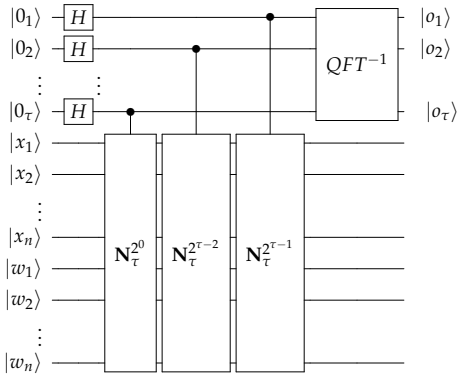


Figure 5: Quantum perceptron circuit proposed in [7].

In [13], a quantum neuron which executes any nonlinear function is proposed. It considers a quantum neuron with n inputs $x_1, x_2, ..., x_n$ and each input having a correspondent weight $w_1, w_2, ..., w_n$. The inputs and weights have p and q qubits respectively. The activation function, $f(x_1, x_2, ..., x_n, w_1, w_2, ...w_n)$ and propagation function, $g(f(x_1, x_2, ..., x_n, w_1, w_2, ...w_n))$, is executed in the phase of quantum state. This quantum neuron has the quantum operator $\mathbf{N}_\tau$ that is shown in Equation 27. The τ value is the precision of the neuron output. It is the amount of qubits of the state where the result of quantum neuron will be loaded. The circuit of this quantum neuron is shown in Figure 7.

$$\mathbf{N}_\tau = \sum_{x_1=0}^{2^m-1} ... \sum_{x_n=0}^{2^m-1} \sum_{w_1=0}^{2^m-1} ... \sum_{w_n=0}^{2^m-1} e^{\frac{2\pi i}{2^\tau} g(f(x_1,...,x_n,w_1,...,w_n))} |x_1, ..., x_n, w_1, ..., w_n\rangle\langle x_1, ..., x_n, w_1, ..., w_n| \tag{27}$$



s

Figure 6: Quantum neuron architecture with any activation functions implementation. The operator of the neuron $\mathbf{N}_\tau$ executes the nonlinear function of the neuron in function of the inputs $|x_1, ..x_n\rangle$ and weights $|w_1, ...w_n\rangle$. The result is on the first quantum register phase through successive τ Controlled-NOT operations. The $\mathbf{QFT^{-1}}$ operator recovers this phase information and transforms it in the quantum state $|o_1, ..., o_\tau\rangle$.

In [43], the brain is analysed as a quantum system. This model points out that the neuron can include an internal dynamics. It considers that the process of recognition changes the internal content of the neuron. This

internal modification is viewed as a memory space which is dynamical. In [3], it is shown that internal state in a neuron induces a chaotic dynamics and simulates more realistic the working of the biologic neuron. In [14], we discuss the internal memory modification as a feature of a quantum perceptron. Then we included in the quantum perceptron proposed in [7] the internal memory and we discussed how this modification affects the operation of the neuron. Three approaches were proposed to represent the quantum perceptron with internal memory. In the first case, the internal state memory is changed by the output generated by the perceptron. In the second case, the internal state can be changed by unitary operators without auxiliary qubits. In that case, the functions that modify the internal memory are characterised to be only the reversible functions. And, in the third case, the internal state is modified by an arbitrary non-reversible boolean function via any unitary operator.

In this paper, we show a quantum neuron model which includes the internal memory and can interact with environment coupled with nonlinear operators.

4. Quantum generalised neuron model with internal memory

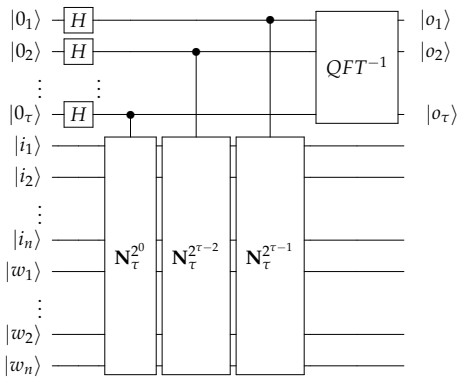


Figure 7: Quantum generalized neuron.

The quantum neuron with internal memory are here in 3 types: the first model considers that the output of neuron is the input in the next iteration of the neuron. The second model considers that the internal memory is altered by a reversible operator in function of the input, weights and internal memory. In third model the internal memory is altered by an arbitrary function in each iteration in function of the input, weights and internal memory. Each type of neuron can be embedded with nonlinear operator which represent the interaction of the neuron with the environment. To represent the quantum neuron we use in the quantum circuit the operator $\mathbf{N}$.

4.1. Quantum neuron output-as-memory

The simplest case of the neuron with internal memory is the quantum neuron whose internal memory is updated in each iteration and its value is the output of the neuron. The circuit of this type of neuron is in Figure 8. This kind of neuron has low memory. Its working takes account only the information of the last neuron iteration.

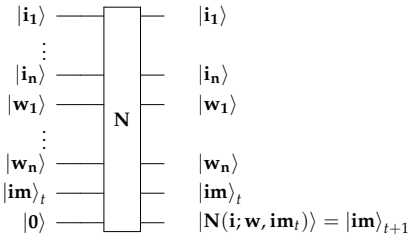


Figure 8: Quantum circuit of the quantum genralised neuron of type A. This model considers that the internal state of the neuron is the neuron output in the last iteration.

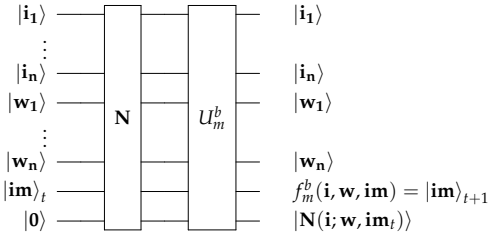


Figure 9: Quantum circuit of the quantum generlised neuron of type B. This model considers that the internal state of the neuron is altered by a quantum oeprator with no ancillary qubits.

4.2. Quantum neuron memory updating

In this model, we modify the internal memory by a quantum reversible operator. It is shown in Figure 9 by the U_m^b operator. This operator does not use ancilla qubits and produces only reversible operators in the internal state. This neuron has a limited set of modification because no ancilla qubits are considered in this process. This can be viewed as a neuron with reduced available memory and low interaction with other neurons.

4.3. Quantum neuron output-as-memory

In the third case, the quantum neuron can changes its content by any quantum operator, using ancilla qubits. This operation is universal and can transform the internal memory by any transformation. The circuit of the type of quantum neuron is shown in Figure 10.

4.4. General quantum neuron with dynamical memory

One can represent the general quantum circuit which represents the quantum neuron with internal memory in Figure 11. For $n+1$ inputs, we have n weights and one internal memory state which is modified during its iteration depending on the type of the neuron.

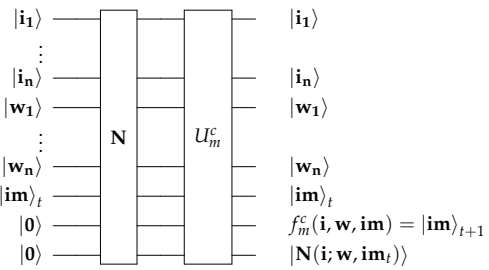


Figure 10: Quantum circuit of the quantum generalised neuron of type C. This model considers that the internal state of the neuron is altered by any quantum operators with either ancillary qubits or not.

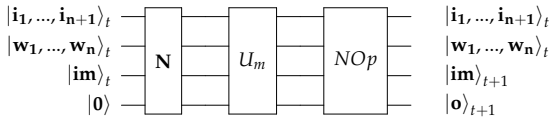


Figure 11: General representation of the quantum neuron with internal memory. **N** operator is the unitary operator which interacts input, neuron weights, neuron internal state and output. U_m operator is the operator which modifies the internal memory depeding on the type of the neuron. The NOp operator is the nonunitary operator which simulates the environment interaction with the neuron. The qubit state $|\mathbf{im}\rangle$ is the internal quantum state and $|\mathbf{o}\rangle$ is the quantum output state.

4.5. *Analytical analysis*

To see the impact to join a quantum operator with quantum nonlinear operators, we set this experiment. This experiment shows the parametric sensitivity and bifurcation during their dynamics of nonlinear quantum operators in time. To encounter the formula of this dynamics, we use the method proposed in [32]. First a nonunitary quantum operator is applied to a quantum bit. After that a quantum operator is applied. The general circuit of this operation is shown in Figure 12. To simplify the experiment we choose here a quantum unitary rotation operator to be applied during this operation. This operator is parametric to generate any rotation, as described in the following:

$$U(x,\phi) = \begin{pmatrix} cos(x) & sin(x)e^{i\phi} \\ -sin(x)e^{-i\phi} & cos(x) \end{pmatrix} \tag{28}$$

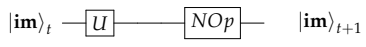


Figure 12: Quantum neuron with internal memory with one qubit embedded with a nonlinear operator.

As the initial input qubit is in a specific shape, the qubit after these transformation is analyzed from the same initial shape. The variation of the amplitudes of the qubit represents the dynamics transformation. In [32], the initial structure of the qubit is $N(z|0\rangle + |1\rangle)$ (we call this set in the experiment as F_1 function, where here N is

the normalization factor. Here we also use the initial input state as $N(z|0\rangle + (1-z)|1\rangle)$ (we call this one as F_2 function), to generate more states than the first shape allows. Then the dynamics process is verified as

$$|\psi'\rangle = US|\psi\rangle \tag{29}$$

where S is a nonunitary operator and U is the unitary operator described in 28.

For each configuration of S, nonlinear operator configuration, x_0 initial input and p dynamical parameter, we generate an orbital diagram. Orbital diagram is a graphic where we plot the values of the dynamics after the dynamics transient for different dynamics parameter.

4.5.1. Unitary evolution

We set in this experiment the behaviour when there is only linear operators in the dynamics. For a given input state $|\psi\rangle = N(z|0\rangle + |1\rangle)$, after the unitary operation, the qubit is

$$U|\psi\rangle = (N.z.cos(x) + N.sin(x)e^{i\phi})|0\rangle + (-N.z.sin(x)e^{-i\phi} + N.cos(x))|1\rangle \tag{30}$$

The unitary evolution is the application of the only unitary operator. Then, the evolution can be described as

$$z' = \frac{N.cos(x) + N.sin(x)e^{i\phi}}{-N.z.sin(x)e^{-i\phi} + N.cos(x)} \tag{31}$$

or

$$z' = \frac{z+p}{-z.p^* + 1} \tag{32}$$

where $p = tan(x)e^{-i\phi}$ and p^* is the complex conjugate value of p.

Considering the input state given by $|\psi\rangle = N(z|0\rangle + (1-z)|1\rangle)$, the application of unitary operation produces following qubit

$$U|\psi\rangle = (N.z.cos(x) + N.(1-z).sin(x)e^{i\phi})|0\rangle + (-N.z.sin(x)e^{-i\phi} + N.(1-z).cos(x))|1\rangle \tag{33}$$

Then the dynamics can be represented by

$$z' = \frac{z + (1-z).p}{-z.p^* + (1-z)} \tag{34}$$

where $p = tan(x)e^{-i\phi}$ and p^* is the complex conjugate value of p.

The orbital diagrams of these dynamics for two initial inputs and two different parameters are plotted in Figure 17.

4.5.2. Bechmann et. al. operator

For a given input state $|\psi\rangle = N(z|0\rangle + |1\rangle)$, the application of the Bechmann *et. al.* operator in this qubit generates this following qubit

$$S|\phi\rangle = N(z^2|0\rangle + |1\rangle) \tag{35}$$

Applying the unitary rotation operator after this transformation generates this qubit:

$$US|\psi\rangle = (N.z^2.cos(x) + N.sin(x)e^{i\phi})|0\rangle + (-N.z^2.sin(x)e^{-i\phi} + N.cos(x))|1\rangle \tag{36}$$

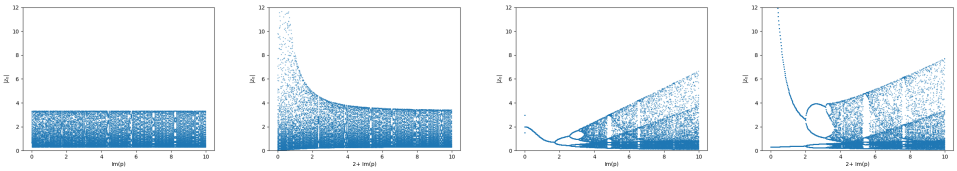


Figure 13: $x_0 = 0.3$ and p varies 0 to 10. for the F_1 function.

Figure 14: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 15: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 16: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 17: Orbital diagrams of the dynamics of the quantum operator applied to unitary operator for the two cases of dynamical functions.

The dynamics is then described as:

$$z' = \frac{z^2 + p}{-z^2.p^* + 1} \tag{37}$$

where $p = tan(x)e^{-i\phi}$ and p^* is the complex conjugate value of p.

Considering the initial input state as $|\psi\rangle = N(z|0\rangle + (1-z)|1\rangle)$, The transformation for a unique qubit is:

$$S|\phi\rangle = N(z^2|0\rangle + (1-z)^2|1\rangle) \tag{38}$$

After the unitary operator application, the output qubit is

$$US|\psi\rangle = (N.z^2.cos(x) + N.(1-z)^2.sin(x)e^{i\phi})|0\rangle + (-N.z^2.sin(x)e^{-i\phi} + N.(1-z)^2.cos(x))|1\rangle \tag{39}$$

The final dynamics is described as:

$$z' = \frac{z^2 + (1-z)^2.p}{-z^2.p^* + (1-z)^2} \tag{40}$$

where $p = tan(x)e^{-i\phi}$ and p^* is the complex conjugate value of p.

The orbital diagrams of these dynamics of two initial inputs and two different parameters are plotted in Figure 22.

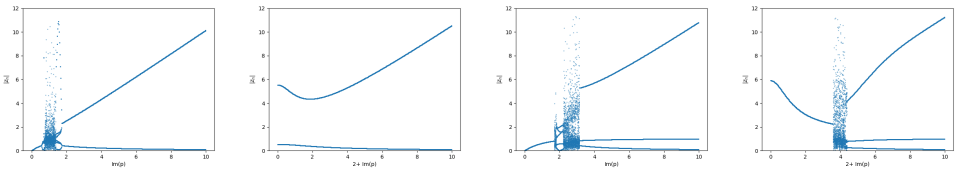


Figure 18: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 19: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 20: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 21: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 22: Orbital diagrams of the dynamics of the Bechmann *et. al* nonlinear operator applied to unitary operator for the two cases of dynamical functions.

4.5.3. *Leporati et. al. operator*

In our experiments, we consider the Leporati *et. al.* operator as two possible operators, *i.e.* $L_1 = 2^n|1\rangle\langle 1|$ and $L_2 = 2^n|0\rangle\langle 0|$.

For a given $|\psi\rangle = N(z|0\rangle + |1\rangle)$, for L_1 Leporati operator,

$$L_1|\psi\rangle = 2^n.N|1\rangle) \tag{41}$$

Then, the transformation after the unitary operation is described as

$$UL|\psi\rangle = 2^n(N.sin(x)e^{i\phi})|0\rangle + 2^n(N.cos(x))|1\rangle \tag{42}$$

The final dynamics formula is described as follows

$$z' = \frac{sin(x)e^{i\phi}}{cos(x)} = p \tag{43}$$

For a given input quantum state $|\psi\rangle = N(z|0\rangle + (1-z)|1\rangle)$, for L_1 Leporati operator, the application of the nonunitary operation generates this following transformation

$$L|\psi\rangle = 2^n.N.(1-z)|1\rangle) \tag{44}$$

Aplying after this, the unitary rotation operator, the qubit transformation is described as

$$UL|psi\rangle = 2^n(N.(1-z)sin(x)e^{i\phi})|0\rangle + 2^n(N.(1-z).cos(x))|1\rangle \tag{45}$$

The final dynamics formula is then

$$z' = \frac{sin(x)e^{i\phi}}{cos(x)} = p \tag{46}$$

Considering the initial input state $|\psi\rangle = N(z|0\rangle + |1\rangle)$, after the application of the L_2 Leporati *et al* operator, the qubit is

$$L_2|\psi\rangle = 2^n(N.z)|0\rangle \tag{47}$$

Then the unitary rotation operator is applied and generates this following qubit

$$UL_2|\psi\rangle = 2^n(N.z.cos(x))|0\rangle - 2^n(N.z.sin(x).e^{-i\phi})|1\rangle \tag{48}$$

The final output dynamics formula is

$$z' = -\frac{1}{p^*} \tag{49}$$

where $p = tan(x)e^{i\phi}$.

For a given input state $|\psi\rangle = N(z|0\rangle + (1-z)|1\rangle)$ the L_2 Leporati et al operator generates this following qubit

$$L_2|\psi\rangle = 2^n(N.z)|0\rangle \tag{50}$$

The unitary rotation operator is applied and generates this following qubit

$$UL_2|psi\rangle = 2^n(N.z.cos(x))|0\rangle - 2^n(N.z.sin(x)e^{-i\phi})|1\rangle \tag{51}$$

The final dynamics formula is described as

$$z' = -\frac{1}{p^*} \tag{52}$$

where $p = tan(x)e^{i\phi}$.

The orbital diagrams of these dynamics of two initial inputs and two different parameters are plotted in Figure 27. One notes that the operator L_1 is the same formula for the F_1 and F_2 input configuration. The same is encountered to the L_2 operator.

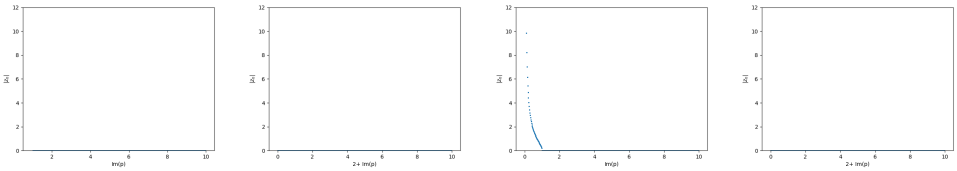


Figure 23: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 24: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 25: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 26: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 27: Orbital diagrams of the dynamics of the Leporati *et. al* nonlinear operators applied to unitary operator for the two cases of dynamical functions.

4.5.4. Lloyd et. al. operator

The Lloyd *et. al.* operator is described to be applied in the quantum state with more than one qubit. In our experiment, we use the inspiration of the LLoyd *et al* operator for one qubit keeping its logic operation.

For a given input state $|\psi\rangle = N(z|0\rangle + |1\rangle)$, the application of the LLoyd *et al* operator for one qubit is described as

$$L_l|\psi\rangle = \begin{cases} |0\rangle, & \text{if} |\psi\rangle = |0\rangle. \\ |1\rangle, & \text{otherwise.} \end{cases} \tag{53}$$

This inspiration is plausible because for two qubits one can create the operator $\mathbf{I} \otimes \mathbf{L_l}$ that does not change the first qubit and applies the Lloyd *et. al.* operator in the second one. There is no configuration that transforms $|\psi\rangle = |0\rangle$, because $z \in [0, 1]$. the nonunitary operator application generates always $L_l\psi = |1\rangle$. Then $L_l|\psi\rangle = |0\rangle$, for any $|\psi\rangle$. After the unitary rotation operator, the qubit is as following

$$UL_l|\psi\rangle = (sin(x)e^{i\phi})|0\rangle + cos(x)|1\rangle \tag{54}$$

And the final dynamics state is

$$z' = \frac{sin(x)e^{i\phi}}{cos(x)} = p \tag{55}$$

For a given input state $|\psi\rangle = N(z|0\rangle + (1-z)|1\rangle)$, the application of the LLoyd *et. al.* operator is the same:

$$L_l|\psi\rangle = \begin{cases} |0\rangle, & \text{if} |\psi\rangle = |0\rangle. \\ |1\rangle, & \text{otherwise.} \end{cases} \tag{56}$$

But, in this case, if $z = 1.0$, $|\psi\rangle = |0\rangle$ and then $L_l|\psi\rangle = |0\rangle$. Then, the output qubit is as follows for $z \neq 1$.

$$UL_l|\psi\rangle = (cos(x))|0\rangle + (-sin(x)e^{-i\phi})|1\rangle \tag{57}$$

Then final dynamics formula is described as follows

$$z' = \frac{-1}{p^*} \tag{58}$$

where $p = tan(x)e^{i\phi}$.

In the case that $z \neq 1.0$ then $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\beta \neq 0$. The formula after the unitary rotation operator is

$$UL_l|\psi\rangle = (sin(x)e^{i\phi})|0\rangle + (cos(x))|1\rangle \tag{59}$$

and the final dynamics formula is described as follows

$$z' = \frac{sin(x)e^{i\phi}}{cos(x)} = p \tag{60}$$

The orbital diagrams of these dynamics of two initial inputs and two different parameters are plotted in Figure 32.

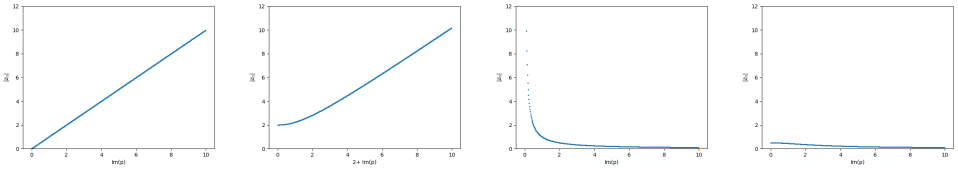


Figure 28: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 29: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 30: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 31: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 32: Orbital diagrams of the dynamics of the Lloyd *et. al* nonlinear operator applied to unitary operator for the two cases of dynamical functions.

4.5.5. Quadratic measurement operator

Considering the input state in the shape $|\psi\rangle = N(z|0\rangle + |1\rangle)$, after the application of the quadratic measurement operator, the qubit is described as follows

$$S|\psi\rangle = N(|z||0\rangle + |1\rangle) \tag{61}$$

After the unitary rotation operator, the qubit is

$$U(S|\psi\rangle) = \frac{N.|z|.cos(x) + N.sin(x)e^{i\phi}}{-N.|z|.sin(x)e^{-i\phi} + N.cos(x)} \tag{62}$$

Then, the final dynamics formula is described as follows

$$z' = \frac{|z| + p}{-|z|.p^* + 1} \tag{63}$$

where $p = tan(x)e^{i\phi}$.

For a given input state $|\psi\rangle = N(z|0\rangle + (1-z)|1\rangle)$, the application of the nonunitary operator leads to the qubit in the following configuration

$$S|\psi\rangle = N(|z||0\rangle + |1-z||1\rangle) \tag{64}$$

After the unitary rotation operator, the output qubit is

$$U(S|\psi\rangle) = \frac{N.|z|.cos(x) + N.|1-z|.sin(x)e^{i\phi}}{-N.|z|.sin(x)e^{-i\phi} + N.|1-z|.cos(x)} \tag{65}$$

Then the final dynamics formula is

$$z' = \frac{|z| + |1 - z|p}{-|z|.p^* + |1 - z|} \tag{66}$$

where $p = tan(x)e^{i\phi}$.

The orbital diagrams of these dynamics of two initial inputs and two different parameters are plotted in Figure 37.

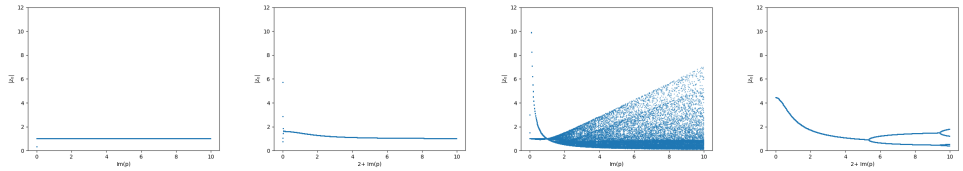


Figure 33: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 34: $x_0 = 0.3$ and p varies 0 to 10 for the F_1 function.

Figure 35: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 36: $x_0 = 0.3$ and p varies 0 to 10 for the F_2 function.

Figure 37: Orbital diagrams of the dynamics of the Quadratic Measurement nonlinear operator applied to unitary operator for the two cases of dynamical functions.

Through these experiments of orbital diagrams, one can see the powerfull of embedding a quantum neuron with a nonlinear operator with a predefined input configuration. We plot the graphs of the Lyapunov exponent of these dynamics in Figure 45. We calculate the derivative of the functions using the Mathematica software [44] and the Lyapunov Exponent approximation defined in [45]. We see that the Lyapunov exponent is positive in unitary operator dynamics, in Bechmann *et. al.* in F_2 function dynamics and in Quadratic Measurement dynamics in F_2 function dynamics. As it is mentioned in [32], quantum coupled quantum system with nonlinear operators in a set of parametrized configuration are able to present chaotic behaviour [32, 46, 23]. This behaviour is encountered in biologic system cells and it is expected that artificial models can mimic behaviour of that biologic models to be complex enough to develop also complex task. The unitary evolution was also studied in this experiment to use as a reference. We see the bifurcation and chaotic behaviour in the orbital diagrams plots. This means that the the dynamics of the quantum operator can be enriched depending on way the input is configured. The Bechmann *et. al.* and the Quadratic measurement operators shows bifurcation and chaotic behavior while we do not encounter complexity during the dynamics of the nonlinear operators of Lloyd *et. al.* and Leporati *et. al.*

4.6. Training and parameter fitting

The process of training the neuron weights to fit for some training dataset can use the intrinsic paralelism feature. This can be done using existing quantum training algorithms [47, 34]. The process to search the best architecture for the quantum neural network can be used as is shown in [48]. An example of quantum architecture of three layers of quantum neurons with internal memory is shown in Figure 46. For a given problem, the space of input can be sliced to 4 quantum neurons and the neurons outputs is the input for the next neurons in other layer.

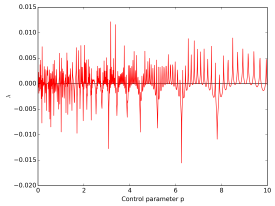


Figure 38: for unitary dynamics and F_1 function.

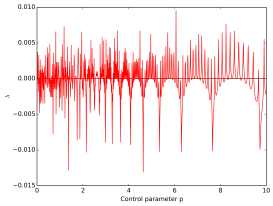


Figure 39: for unitary dynamics and F_2 function.

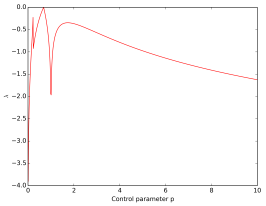


Figure 40: for Bechmann *et. al.* dynamics and F_1 function.

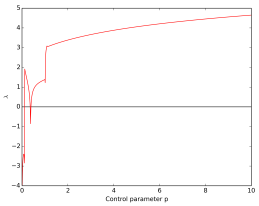


Figure 41: for Bechmann *et. al.* dynamics and F_2 function.

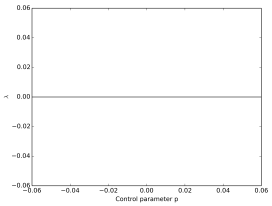


Figure 42: for Leporati *et. al.* and Lloyd *et. al* dynamics and F_1 and F_2 functions.

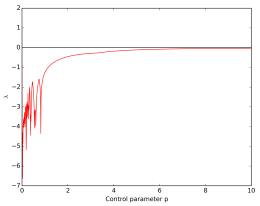


Figure 43: for Quadratic Measurement dynamics and F_1 function.

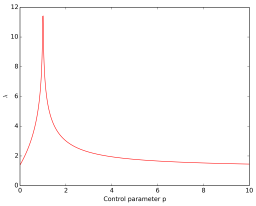


Figure 44: for Quadratic Measurement dynamics and F_2 function.

Figure 45: Lyapunov exponent graphs for the different non-unitary operators dynamics.

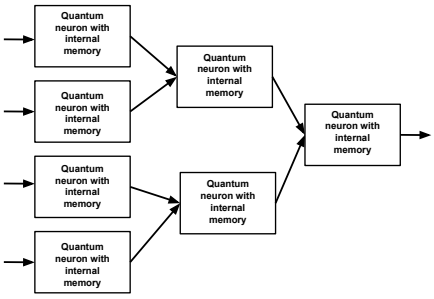


Figure 46: Three layers of quantum neurons with internal memory.

5. Conclusion

In this paper, we presented a generalised architecture for weighted quantum neuron which includes in its working a dynamical internal memory. This generalisation allows the quantum neuron to variate its internal memory in each execution of the neuron. The weights and activate and propagation functions are parameters of the neuron which can be modified by a training process. The quantum inverse Fourier transform operator is used to calculate the activate and propagation function depending on input and weight values. The modification of internal memory was presented as 3 possible neuron models. The first model of neuron modifies the internal memory in function of the output of the neuron. This type has low memory and only saves the last output. The second model modifies the internal memory considering the input, weights and internal memory as its variables using quantum reversible operators. This model considers low interaction with other neurons and low memory. In the third model, any quantum operator can change the internal memory content. These three types of neuron models include the three possibilities to change a content of a quantum state. This is a generalisation of the perceptron quantum neuron presented in [7] and [14]. In [7], the perceptron model is dependent of the input and weights but it cannot work as expected if the inner product is negative. In [23], this perceptron model was embedded with a dynamical internal memory but works only for step function. With a dynamical internal memory, the neuron can consider to calculate temporal information and to solve a more set of problems [3]. Dynamic neural networks gives the possibility of retaining information to be used in other executions. In contrast to static neural networks, without memory or recurrence, the dynamical neural network can generate diverse output vectors in response to the same input vector, because the response may also depend on the actual state of the existing memories [49].

We also presented the impact of embedding, in a quantum operator, linear and nonlinear quantum operators. The Lyapunov Exponent graphs are drawn and we pointed the parameter values which invoke chaos in the dynamics. We encountered chaotic behaviour using linear operators in a specific input shape. Bifurcation and chaotic behaviour is viewed. We extended a dynamical experiment firstly proposed by [32]. As it was shown in previous works, neuron cells present chaotic and nonlinear operators [3] then we proposed this quantum neuron which has possibility to mimic the biological neuron. Nonlinear quantum operators have been used in quantum computing to solve fast some problems in quantum computing probabilistically [27].

References

[1] X. Y. Zhang, F. Yin, Y. M. Zhang, C. L. Liu, Y. Bengio, Drawing and recognizing chinese characters with recurrent neural network, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 849–862.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Scientific Reports 8 (1) (2018) 6085.

[3] K. Aihara, T. Takabe, M. Toyoda, Chaotic neural networks, Physics Letters A 144 (6) (1990) 333 – 340.

[4] Y. Hirata, M. Oku, K. Aihara, Chaos in neurons and its application: Perspective of chaos engineering, Chaos: An Interdisciplinary Journal of Nonlinear Science 22 (4) (2012) 047511.

[5] A. K. Tiba, A. F. Araujo, M. N. Rabelo, Hopf bifurcation in a chaotic associative memory, Neurocomputing 152 (2015) 109 – 120.

[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (2017) 195 EP –.

[7] M. Schuld, I. Sinayskiy, F. Petruccione, Simulating a perceptron on a quantum computer, Physics Letters A 379 (7) (2015) 660 – 663.

[8] W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, Vector space weightless neural networks, in: ESANN 2014, no. April, 2014, pp. 535–540.

[9] M. V. Altaisky, Quantum neural network, Technical report, Joint Institute for Nuclear Research, Russia (2001).

[10] D. Ventura, T. Martinez, Quantum associative memory, Information Sciences 124 (1–4) (2000) 273 – 296.

[11] E. C. Behrman, J. Niemel, J. E. Steck, S. R. Skinner, A quantum dot neural network, in: Proceedings of the 4th Workshop on Physics of Computation, Citeseer, 1996, pp. 22–24.

[12] A. J. da Silva, T. B. Ludermir, W. R. de Oliveira, Quantum perceptron over a field and neural network architecture selection in a quantum computer, Neural Networks 76 (2016) 55 – 64.

[13] F. M. de Paula Neto, T. B. Ludermir, W. R. de Oliveira, A. J. a. da Silva, Implementing nonlinear activation functions in aquantum neuron, submitted.

[14] F. M. de Paula Neto, T. B. Ludermir, W. R. de Oliveira, , A. J. da Silva, Quantum perceptron with dynamic internal memory, in: 2018 International Joint Conference on Neural Networks, IJCNN 2018, IEEE, 2018.

[15] D. S. Abrams, S. Lloyd, Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems, Phys. Rev. Lett. 81 (18) (1998) 3992–3995.

[16] H. Terashima, M. Ueda, Nonunitary quantum circuit, eprint arXiv:quant-ph/0304061arXiv:quant-ph/0304061.

[17] C. Williams, R. Gingrich, Non-unitary probabilistic quantum computing circuit and method, uS Patent App. 11/007,792 (Aug. 4 2005).
URL http://www.google.com/patents/US20050167658

[18] A. Leporati, S. Felloni, Three "quantum" algorithms to solve 3-sat, Theoretical Computer Science 372 (2–3) (2007) 218 – 241, membrane Computing.

[19] A. Leporati, C. Zandron, G. Mauri, Simulating the fredkin gate with energy-based p systems., J. UCS 10 (5) (2004) 600–619.

[20] M. Ohya, I. Volovich, Quantum Computing, NP-complete Problems and Chaotic Dynamics, eprint arXiv:quant-ph/9912100arXiv:quant-ph/9912100.

[21] F. M. de Paula Neto, T. B. Ludermir, W. R. De Oliveira, A. J. da Silva, Solving np-complete problems using quantum weightless neuron nodes, in: Intelligent Systems, 2015 Brazilian Conference on, 2015, pp. 258–263.

[22] M. Panella, G. Martinelli, Neural networks with quantum architecture and quantum learning, International Journal of Circuit Theory and Applications 39 (1) (2011) 61–77.

[23] F. M. de Paula Neto, W. R. de Oliveira, T. B. Ludermir, A. J. da Silva, Chaos in a quantum neuron: An open system approach, Neurocomputing 246 (2017) 3 – 11.

[24] R. M. Gingrich, C. P. Williams, Non-unitary probabilistic quantum computing, in: Proceedings of the Winter International Synposium on Information and Communication Technologies, WISICT '04, Trinity College Dublin, 2004, pp. 1–6.

[25] H. Terashima, M. Ueda, Nonunitary quantum circuit, International Journal of Quantum Information 3 (04) (2005) 633–647.

[26] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.

[27] D. S. Abrams, S. Lloyd, Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems, Phys. Rev. Lett. 81 (18) (1998) 3992–3995.

[28] R. Zhou, H. Wang, Q. Wu, Y. Shi, Quantum associative neural network with nonlinear search algorithm, International Journal of Theoretical Physics 51 (3) (2012) 705–723.

[29] J.-P. Tchapet Njafa, S. Nana Engo, Concise quantum associative memories with nonlinear search algorithm, Fortschritte der Physik 64 (2-3) (2016) 250–268.

[30] M. Ohya, I. V. Volovich, Quantum computing, np-complete problems and chaotic dynamics (1999).

[31] H. Bechmann-Pasquinucci, B. Huttner, N. Gisin, Non-linear quantum state transformation of spin-1/2, Physics Letters A 242 (1998) 198–204.

[32] T. Kiss, I. Jex, G. Alber, S. Vymetal, Complex chaos in the conditional dynamics of qubits, Phys. Rev. A 74 (2006) 040301.

[33] F. M. de Paula Neto, W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, Chaos in Quantum Weightless Neuron Node Dynamics, Neurocomputing 183 (2015) 23–38.

[34] A. J. da Silva, W. R. de Oliveira, T. B. Ludermir, Classical and superposed learning for quantum weightless neural networks, Neurocomputing 75 (1) (2012) 52–60.

[35] R. Zhou, Q. Ding, Quantum m-p neural network, International Journal of Theoretical Physics 46 (12) (2007) 3209–3215.

[36] A. J. da Silva, W. R. de Oliveira, T. B. Ludermir, Comments on "quantum m-p neural network", International Journal of Theoretical Physics 54 (6) (2015) 1878–1881.

[37] M. Panella, G. Martinelli, Neural networks with quantum architecture and quantum learning, International Journal of Circuit Theory and Applications 39 (1) (2011) 61–77.

[38] W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, A. Leonel, W. R. Galindo, J. C. Pereira, Quantum Logical Neural Networks, in: Brazilian Symposium on Neural Networks, 2008, pp. 147–152.

[39] W. R. de Oliveira, Quantum RAM Based Neural Networks, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2009) 22–24.

[40] I. Aleksander, Self-adaptive universal logic circuits, Electronics Letters 2 8 (1966) 321–322.

[41] E. Behrman, L. Nash, J. Steck, V. Chandrashekar, S. Skinner, Simulations of quantum neural networks, Information Sciences 128 (3–4) (2000) 257 – 269.

[42] B. Ricks, D. Ventura, Training a quantum neural network, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems 16, MIT Press, 2004, pp. 1019–1026.

[43] K.-H. Fichtner, K. Inoue, M. Ohya, On a quantum model of brain activities — distribution of the outcomes of eeg-measurements and random point fields, Open Systems & Information Dynamics 19 (04) (2012) 1250025.

[44] W. R. Inc., Mathematica, Version 11.3, champaign, IL, 2018.

[45] S. H. Strogatz, Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity), Westview Press, 1994.

[46] T. Kiss, S. Vymětal, L. Tóth, A. Gábris, I. Jex, G. Alber, Measurement-induced chaos with entangled states, Physical review letters 107 (10) (2011) 100501.

[47] B. Ricks, D. Ventura, Training a quantum neural network, in: Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS'03, MIT Press, Cambridge, MA, USA, 2003, pp. 1019–1026.

[48] A. J. da Silva, R. L. F. de Oliveira, Neural networks architecture evaluation in a quantum computer, in: 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017, pp. 163–168.

[49] L. Medsker, L. Jain, Recurrent neural networks, Design and Applications 5.

APÊNDICE D – NON-UNITARY QUANTUM ASSOCIATIVE MEMORY

2017 Brazilian Conference on Intelligent Systems

Non-unitary Quantum Associative Memory

Fernando M de Paula Neto*, Adenilton J da Silva†, Wilson R de Oliveira†, Teresa B Ludermir*
*Centro de Informática
Universidade Federal de Pernambuco, Recife, Cidade Universitária, 50670-901
Email: {fmpn2, tbl}@cin.ufpe.br
†Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco, Recife, Dois Irmãos, 52171-900
Email: adenilton.silva@ufrpe.br, wilson.rosa@gmail.com

***Abstract*—Quantum associative memories (QAM) have storage capacity exponentially larger than classical associative memories. Given an input with n bits, the QAM can load all of 2^n binary patterns. In this paper we propose a non-unitary quantum memory retrieval algorithm. The proposed method is used to recover incomplete input information and also in pattern classification task as an ensemble of memories. The advantages of the proposed method are the simplicity of recovery procedure in a unique iteration and the fact the circuit has linear complexity in function of the input length with deterministic output.**

I. Introduction

Various algorithms have been proposed to simulate the brain activity of recovering information stored in the memory. Sometimes that input information is unrelated or incomplete in comparison with the respectively saved one. This kind of model is called in neural networks field as associative memory. The most well known model is the Hopfield network [1] and bidirectional associative memory (BAM) [2]. A limitation of these classical models is that we are only able to store m patterns, using n neurons, with $m \leq kn$ and $0.15 \leq k \leq 0.5$.

Recently it has been proposed the use of quantum computing principles to create associative memories with exponential memory capacity. In [3], a quantum associative memory is proposed using an adaptation of the Grover's algorithm [4] and using a quantum storing algorithm [5]. In [6] is proposed another quantum storing algorithm with a unitary retrieval algorithm using the Hamiltonian operator which executes the *Hamming distance* between two binary patterns. Quantum weightless neurons are associated with probabilistic quantum memories [6] in [7]. In this model the probabilistic memories give a linear increasing of neuron parameters and training with one single execution of a network of quantum weightless neurons.

A nonlinear operator [8] is used for information retrieval in [9] and uses Binary Superposed Decision Diagrams (BSDD) [10] to store information in the memory. Another nonlinear associative memory is proposed in [11] which claims to simplify the model in [9]. It uses the Weinberg nonlinear operator [12] for informartion retrieval and also uses BSDD to store patterns. The Weinberg operator is used in [13] to solve NP-Complete problems in polynomial time.

Several quantum associative memories have been proposed [14], [6], [15], [11]. The quantum associative memories proposed in [15], [11] performs a nonlinear or nonunitary search in the retrieval procedure. In [11] the retrieval procedure of the nonlinear quantum associative memory has computational cost $O(c - r)$ where $c = log_2 q$, q is the number of stored patterns, $r = int(log_2 m)$ and m is the number of stored patterns that match the search. In [15] the retrieval procedure has cost $O\left(\sqrt{\frac{N}{m}}\right)$ where $N = 2^n$, n is the number of qubits in the memory and m is the number of stored patterns. The quantum memories proposed in [14], [6] use only standard quantum operators. In [14] the retrieval algorithm is based on Grover's search procedure that finds the searched pattern with high probability. And in [6] the retrieval algorithm has computational $O(N)$ but its output is probabilistic and depends on the stored data distribution. In [16], a nonunitary retrieval procedure is proposed to recover contents in a quantum associative memory using a nonunitary operator which is selected depending of the input information to be recovered.

The objective of this paper is to propose a nonunitary quantum associative memory with retrieval algorithm that combines the high probability output of [14] and the computational cost of [6]. To achieve this objective we use the non-unitary Leporati operator [17] to recover the patterns of a quantum memory. Using this non-unitary operator, the outputs have the advantage to be recovered deterministically. Only this nonunitary operator is used to recover the information. The proposed algorithm uses a threshold of similarity to recover a pattern in the memory and it can also be used for partial or incomplete inputs. Non-unitary operators have been used in quantum computing to efficiently solve hard problems such as NP-complete problems [18], [19]. In this paper we also introduce the model proposed in the classification task as an ensemble of memories.

Section II introduces basic concepts of quantum computing. Section III is about quantum memories and the proposed method is presented in Section IV, explaining

978-1-5386-2407-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BRACIS.2017.36

its working for complete or partial input patterns and introducing it in the classification task. Section V contains the conclusions and further directions of work.

II. Quantum Computing

Quantum computing started at least in 1980s with the works of Benioff [20] and Richard Feynman [21]. After that, David Deutsch [22] proposed the quantum Turing machine. A quantum circuit is a mathematical quantisation of the classical Boolean circuit model of computation. Through its intrinsic parallelism (linearity and linear combination), quantum computing can evaluate a function over all of its possible input values. In this Section we present the basic concepts of quantum computing needed in the next Sections. Further details can found in [23].

A. Qubits

The unit of information in quantum computing is the quantum bit, *qubit*. The general state of a qubit is a linear combination (*superposition*) of of elements of the computational basis states $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers and subject to the restriction $|\alpha|^2 + |\beta|^2 = 1$. The computational basis vectors (known in Linear Algebra as the canonical basis), in the Dirac notation, $|0\rangle$ and $|1\rangle$, are the pair of orthonormal vectors $|0\rangle = [1 \quad 0]^T$ and $|1\rangle = [0 \quad 1]^T$.

A superposition of computational-basis states can be also represented as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}.$$

Tensor products are used to represent more than one qubit, or a combination of quantum states, as $|i\rangle \otimes |j\rangle = |ij\rangle$, where $i, j \in \{0, 1\}$. An n-qubit is in the space $\mathbb{C}^{2^n}$. For an n and m dimensional vectors, the tensor products gives a nm-dimensional vector. In this text, we use bold letters to name a superposition of qubits strings, *e.g* $|\mathbf{i}\rangle = \frac{1}{\sqrt{3}}(|1011\rangle + |0100\rangle + |1100\rangle)$, and $|m_k\rangle$ for a specific qubit position k in $|\mathbf{m}\rangle$. Considering *e.g* $|\mathbf{m}\rangle = |010\rangle$, the kth qubits are $|m_0\rangle = |0\rangle$, $|m_1\rangle = |1\rangle$ and $|m_2\rangle = |0\rangle$.

The bra $\langle .|$ of a ket state $|.\rangle$ is its complex conjugate transpose. As example, given a state $|\mathbf{m}\rangle = i|010\rangle$, its bra is $\langle \mathbf{m}| = |\mathbf{m}\rangle^\dagger = -i\langle 010|$.

B. Quantum unitary operators

Closed quantum systems evolve by a unitary operators that change their states. For finite dimensional spaces, as in the case in quantum computing, a quantum operator $\mathbf{U}$ over n qubits is just a $2^n \times 2^n$ complex unitary matrix.

The main unitary one qubit operators used in quantum algorithms are:

- Identity operator $\mathbf{I}$, which leaves the qubit values intact.
$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{array}{l}\mathbf{I}|0\rangle = |0\rangle \\ \mathbf{I}|1\rangle = |1\rangle\end{array}$$
- The flip operator, $\mathbf{X}$, behaves as the classical NOT, changing where the state is $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$.
$$\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{array}{l}\mathbf{X}|0\rangle = |1\rangle \\ \mathbf{X}|1\rangle = |0\rangle\end{array}$$
- Hadamard operator, $\mathbf{H}$, generates a balanced (equally probable) superposition of states. This operator is commonly used to generate a sequence of all possible values of a string of qubits.
$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \begin{array}{l}\mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle)\end{array}$$

In the same way we can combine quantum states, quantum operators can also be combined using tensor product. For two (n_0, m_0)-dimensional matrix U and (n_1, m_1)-dimensional matrix V, their composition, $U \otimes V$, products a third $(n_0 n_1, m_0 m_1)$-dimensional matrix. We denote as $\mathbf{A}^{\otimes s}$ the s-fold application of $\mathbf{A}$.

The **CNOT** is a two qubits operator. It has a control qubit and a target qubit. It works considering the value of the control qubit to apply the $\mathbf{X}$ operator on the target qubit. If the control qubit is set to 1 the $\mathbf{X}$ operator is applied to target qubit. The matrix representation for in the computational basis is shown in Equation 1.

$$\mathbf{CNOT} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix} \quad \begin{array}{l}\mathbf{CNOT}|00\rangle = |00\rangle \\ \mathbf{CNOT}|01\rangle = |01\rangle \\ \mathbf{CNOT}|10\rangle = |11\rangle \\ \mathbf{CNOT}|11\rangle = |10\rangle\end{array} \tag{1}$$

We can generalise and define a $(n+1)-$ary **CNOT** having n control qubits and requring all the control qubits to be $|1\rangle$ for applying $\mathbf{X}$.

The representation of quantum circuits is done considering the qubit as a wire and the operator as a box. The flow of the execution, as in the classical case, is from left to right.

Figure 1 has an example of a quantum circuit composed of a **CNOT**, where the control qubit is depicted by a filled circle, and a **H** operator.

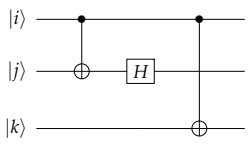


Fig. 1. An example of quantum circuit with two **CNOT** operators and a **H** operator.

C. Non-unitary quantum operators

Measurement is an irreversible operation that partial or totally loses the information about superposition of states. For a qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, a measurement

collapses (projects) the state either to $|0\rangle$ state with $|\alpha|^2$ of probability or to $|0\rangle$ with $|\beta|^2$ probability. For a composed and superposed quantum state $|\mathbf{g}\rangle$ a probability to see a state $|\mathbf{i}\rangle$ is $|\langle g|i\rangle|^2$. In Figure 2 an example of the measurement operator in a quantum circuit is shown.

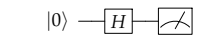


Fig. 2. Measurement results 0 or 1 with equal probability $\frac{1}{2}$.

Other non-unitary operators have been proposed and their effects in quantum computation analysed. Non-unitary and non-linear operators [24][25][26][18] combined with quantum gates solve analytically NP-complete problems in polynomial time [17][27][19]. Quantum neural networks can be trained by non-linear operators [28], [29]. Nonlinear effects in quantum dynamics as chaos [30] can be observed when non-unitary operators are used. Although there are controversies about the physical realisation of non-unitary operators, in [26], [31] a quantum unitary operator is embedded in a unitary operator and can be applied probabilistically. It is shown in [32] that the non-unitary operators can be considered as universal set of gates.

In this paper we use a non-unitary operator first defined in [18] which we call the Leporati operator. The Leporati operator is described as follows

$$O_f = 2^n \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = 2^n |1\rangle\langle 1|. \tag{2}$$

This operator without the amplitude amplification, *i.e.* $|1\rangle\langle 1|$, was also used in [33] for a non-linear quantum state transformation to distinguish optimally two non-orthogonal states.

In essence, the Leporati operator outputs a quantum state with the amplitude of the $|1\rangle$ amplified by 2^n and erase the amplitude of $|0\rangle$. In the basis states it behaves as

$$O_f|0\rangle = \mathbf{0}$$

and

$$O_f|1\rangle = 2^n|1\rangle$$

for a given n. In a superposition state the operator behaves as

$$O_f(\alpha|0\rangle + \beta|1\rangle) = 2^n\beta|1\rangle.$$

III. Quantum memories

Quantum Associative memories have exponential memory capacity in relation to the classical ones. This means that a memory with n bits can store all of 2^n patterns. In [3], a quantum associative memory uses an adaptation of the Grover's algorithm. This model also proposes a pattern storing algorithm [5] which creates a quantum state from a training set for pattern of length n bits using $2n+1$ quantum bits. In the retrieval processing, the Grover's algorithm is adapted to receive a quantum state which is not all the superposition of equally distributed states. The Grover's algorithm is also adapted to receive a partial (incomplete) quantum state as input. There is a probability of success associated with the number of qubits, patterns stored and patterns not stored.

In [6], the quantum storing pattern algorithm uses $2n+2$ qubits and executes a unitary retrieval through the Hamiltonian operator which calculates the *Hamming distance* between two binary patterns. In this retrieval algorithm it is considered the dynamics of the system Hamiltonian. The authors argue that this Hamiltonian is the generalisation of the Hopfield model with efficiently find the exact global minimum of the quantum energy landscape, without the appearance of any spurious memories (*i.e* patterns not desired but possibly included during the training step). Nevertheless this memory is probabilistic, giving an output related with the memory probability distribution. There is no guarantee that an input is recognised and identified correctly.

Our proposed retrieval method for quantum memories needs an initialised quantum memory built by some initialising algorithm. The following subsection introduces and investigates our memory model.

A. Memory initialisation

The initialisation of a memory is an operation that transforms some training set into a superposed quantum state. Given p binary patterns p^i of length n, the memory represented by a quantum state is

$$|m\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} \left|p^i\right\rangle. \tag{3}$$

As we mentioned in the beginning of this Section, there are different algorithms to create a superposition of states which represents a memory. For our purposes we can use two initialisation algorithm proposed by Ventura *et al.* [5] or by Trugenberger *et al.* [6]. They give a quantum state in the shape described in Equation 3. The complexity of these initialising algorithms is linear in function of the number of patterns in the training set..

For instance, if we have a training set as $\mathbf{P} = \{0000, 0010, 1010\}$, the quantum memory associated with this training set after the initialising procedure is

$$|m\rangle = \frac{1}{\sqrt{3}}(|0000\rangle + |0010\rangle + |1010\rangle). \tag{4}$$

IV. Non-unitary quantum memory

The quantum memories differ from each others by the initialisation of the memory and by its recovery method. We consider that a memory can be created using an existing method cited in Section III-A and it is in the shape of Equation 3. After that we propose to use the Leporati operator to encounter a similar content in the

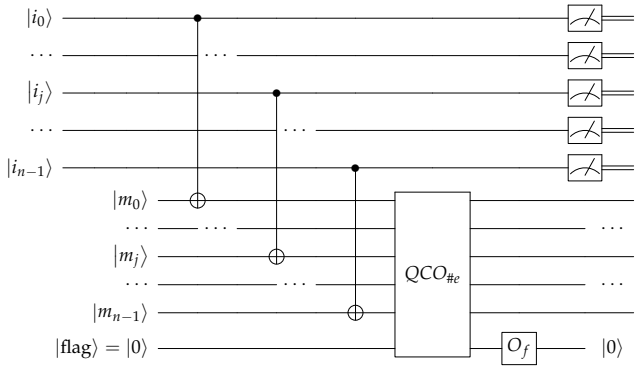


Fig. 3. Quantum circuit of the NuQM. The input qubit state $|i_0 \cdots i_n\rangle$ is the pattern which one wants to check the existence in the memory $|m_0 \cdots m_n\rangle$. The result is encountered in the first n qubits after the measurement. If the pattern exists in the memory, the outputs are the same qubits of the input.

memory for given input pattern. The idea is that the non-unitary makes as a "mask" to cancel the states of the memory which are not similar to the input. We name this memory as Non-unitary Quantum Memory (NuQM).

The retrieval algorithm can be understood in three steps represented in the circuit in Figure 3. The first step is in the **CNOT** gates operation. For a n-qubits pattern, we have n **CNOT** gates in cascade. After this step, the states in memory which have e qubits equal to the input state will have e qubits in state $|0\rangle$. In other words, the state $|\mathbf{i}\rangle|\mathbf{m}\rangle$ after the **CNOT** gates has zeros in the memory state depending on the number of qubits in the memory.

For instance, given a memory $|\mathbf{m}\rangle = \frac{1}{\sqrt{2}}(|0111\rangle + |0011\rangle)$, and an input $|\mathbf{i}\rangle = |0110\rangle$, after the **CNOT** gates, the state will be $|\mathbf{im}\rangle\frac{1}{\sqrt{2}}(|0110\rangle|0001\rangle + |0110\rangle|0101\rangle)$, showing that the first memory state $|0111\rangle$ has 3 qubits equal and the second memory state has 2 qubits equal (by the amount of zero qubits in the final of the string).

The second stage of the recovery method is to check the similarity of the input pattern with the memory patterns. This is done considering a threshold of amount of zeros encountered in the previous step. Any Boolean function can be implemented in a quantum circuit [23]. The $QCO_{\#e}$ operator set 1 in the $|\text{flag}\rangle$ qubit if e or more zeros exist in the memory state after the step 1. This means that the memory recovery states in the memory with at least e qubits equal in relation of the input state. Following the same example of the previous paragraph, after the comparison operator, considering $e = 3$, the states will be $|\mathbf{i}\rangle|\mathbf{m}\rangle|\text{flag}\rangle = \frac{1}{\sqrt{2}}(|0110\rangle|0001\rangle|1\rangle + |0110\rangle|0101\rangle|0\rangle)$.

The last stage of the recovery method is to erase the states which are not set with one in the flag qubit. This is done by the non-unitary Leporati operator, O_f. After this operation, the n first qubits of the memory can be measured indicating the patterns of the memory which are next e qbits from the input state. Considering this non-unitary operation, our method has not probabilistic results, since the search element exists in memory if and only if it appears in the measurement procedure.

As the 2^n exponential factor of the O_f operator is distributed in all states which are not erased, this factor is not repeated in the following examples.

As the measurement is probabilistic in function of amplitude of the recovered states, the case that the memory can retrieve more than one pattern similar to the input pattern, the user needs to execute the memory more than once to check probabilistically all existing memory states. This is done with the memory initialised and repeating the recovery procedure.

We can point out that when an input is not presented in the memory, the measured qubits receive the value of **0** (emptiness). Considering the experiments and analysis undertaken in [18], [19], [26], [32], this aspect of the non-unitary quantum computing is an open-ended question. In this circuit level of implementation, we can check the mathematical result and infer the power of the non-unitary gate but not guaranteeing its physical realisation.

A. Incomplete patterns recovery

For the patterns which are incomplete (*e.g.* in the form $|x00x11x\rangle$, where x is a missing value), we can use the **H** gate to create a superposition of $|0\rangle$ and $|1\rangle$ to replace in the missing qubits positions. In this way we can check in parallel the existence of the all possible state in the memory associated with the missing values.

For instance, for a given $|1xx0\rangle$ pattern with the second and third qubit missing, the new quantum state with $\mathbf{H}|0\rangle = (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle))$ is applied in the second and third qubits, resulting in the quantum state

$$\begin{aligned} |\mathbf{i}\rangle &= |1\rangle(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle))(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle))|0\rangle \\ &= \frac{1}{2}(|1000\rangle + |1010\rangle + |1100\rangle + |1110\rangle). \end{aligned} \tag{5}$$

Considering the memory $|\mathbf{m}\rangle = \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle)$, and using the new state of the Equation 5, we exemplify the process the recovery algorithm. In the first step, after the **CNOT** the state

$$\begin{aligned} |\mathbf{i}\rangle|\mathbf{m}\rangle = \frac{1}{\sqrt{8}}(&|1000\rangle + |1010\rangle + |1100\rangle \\ &+ |1110\rangle)(|0011\rangle + |1100\rangle) \end{aligned} \tag{6}$$

is

$$\begin{aligned} &\frac{1}{\sqrt{8}}(|1000\rangle|1011\rangle|1000\rangle|0100\rangle|1010\rangle|1001\rangle + |1010\rangle|0110\rangle \\ &+ |1100\rangle|1111\rangle + |1100\rangle 0000 + |1110\rangle|1101\rangle \\ &+ |1110\rangle|0010\rangle). \end{aligned} \tag{7}$$

In the second step, considering $e = 4$ (*i.e.* all the memory state to be matched have to be equal to the input), the states $|\mathbf{i}\rangle|\mathbf{m}\rangle|\text{flag}\rangle$ are

$$\begin{aligned} &\frac{1}{\sqrt{8}}(|1000\rangle|1011\rangle|0\rangle|1000\rangle|0100\rangle|0\rangle|1010\rangle|1001\rangle|0\rangle \\ &+ |1010\rangle|0110\rangle|0\rangle + |1100\rangle|1111\rangle|0\rangle + |1100\rangle|0000\rangle|1\rangle \\ &+ |1110\rangle|1101\rangle|0\rangle + |1110\rangle|0010\rangle|0\rangle). \end{aligned} \tag{8}$$

The Leporati operator will cancel out all the states which the flag is not set to 1, resulting in $|1100\rangle|0000\rangle$. After the measurement, the first qubits for this example are 1100.

B. Introducing the NuQM for classification task

The classification task is to detect for a given unknown pattern which group/class it belongs to. The classifier system is trained to detect the group of similar patterns (class) "closer" to a given input pattern using its own criteria. There are different criteria to decide it (more details in pattern classification can be encountered here [34]).

Using our proposed NuQM model, we introduce a classification scheme that has linear space complexity in function of the number of classes.

For a given c classes problem, we can build the memory $|\mathbf{m}^c\rangle$ from the patterns of the class c. Initialising the $|\mathbf{m}^0\rangle, ...|\mathbf{m}^{c-1}\rangle$ memories, we can create a set of c non-unitary memories considering some criteria of similarity e. Figure 4 shows an ensemble with c memories. Given

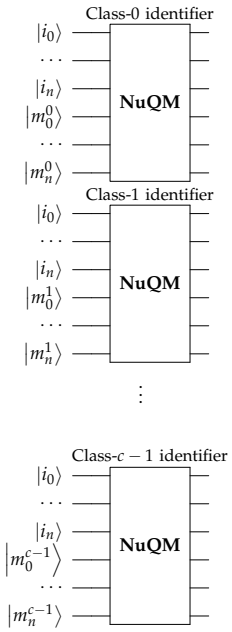


Fig. 4. NuQM Classifier Ensemble formed by c identifiers of the c class. Each identifier is a NuQM which has its $|m^i\rangle$ memory trained by the pattern set of its respectively class.

an unknown binary input $|\mathbf{u}\rangle$, it is tested in each c memory. For a memory i which the pattern is encountered, the classifier outputs the class i.

The problem which the ensemble can encounter is when missing values is in the input. When the superposition of all values of the missing values is set in the input, more than one NuQM identifier can flag the output. This means that the user would choose the class by some other criteria. Future works may be directed to solve this kind of collision.

V. Conclusion

In this paper we described an application of the Leporati non-unitary operator for a quantum memory recovery method. The memory is named as Non-Unitary Quantum Memory (NuQM). The content of the memory can be initialised by some existing initialisation quantum algorithm. In this work we mentioned the Ventura *et al.* [5] and Trugenberger *et al.* [6] algorithms. The simplicity of the circuit and the non-iterative procedure for a quantum memory recovery are its the major advantage. In principle, NuQM recovers the contents deterministically

if the Leporati non-unitary operator is possible to be implemented. Given a memory whose patterns have n qubits, the proposed method takes place in two steps of working using n **CNOT**, a $QCO_{\#e}$ operator, which flag an ancillary qubit to 1 if there is at least e zero qubits in a quantum state, and finally the O_f non-unitary operator. As all the existing quantum memory initialisation procedure have polynomial space complexity in the input length, the NuQM is also working in polynomial space. Using the proposed non-unitary quantum computing algorithm [31], the NuQM can be implemented and checked probabilistically.

The NuQM can be set to tolerate a levels of similarity of the input pattern with the memory content. It can also be applied in problems with missing input values, doing a search in the all possible values of the missing values through quantum parallelism. The task of classification patterns is introduced by an ensemble of NuQMs, showing its working for a c classes problem.

The disadvantage of the proposed method is that the procedure for initialisation of memory may be repeated since the memory is modified in each step of the recovery procedure. In [6], a mechanism to maintain the memory preserved is described. This also can be used to obviate this problem in future works. We are investigating also the use of an algorithm to work with a given distance for the inputs patterns in relation of the memory ones. Further works intend to deal with a unitary version of Leporati non-unitary operator [17].

Acknowledgment

The authors would like to thank CNPq and FACEPE (Brazilian Research Agencies) for their financial support.

References

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[2] B. Kosko, "Bidirectional associative memories," *IEEE Transactions on Systems, man, and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[3] D. Ventura and T. Martinez, "Quantum associative memory," *Information Sciences*, vol. 124, no. 1–4, pp. 273 – 296, 2000.

[4] L. K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997.

[5] D. Ventura and T. Martinez, "Initializing the amplitude distribution of a quantum state," *Foundations of Physics Letters*, vol. 12, no. 6, pp. 547–559, 1999.

[6] C. A. Trugenberger, "Probabilistic Quantum Memories," *Phys. Rev. Lett.*, vol. 87, no. 6, p. 67901, 2001.

[7] A. Silva, W. d. Oliveira, and T. Ludermir, "A weightless neural node based on a probabilistic quantum memory," in *2010 Eleventh Brazilian Symposium on Neural Networks*, Oct 2010, pp. 259–264.

[8] M. Czachor, "Local modification of the abrams-lloyd nonlinear algorithm," *arXiv preprint quant-ph/9803019*, 1998.

[9] R. Zhou, H. Wang, Q. Wu, and Y. Shi, "Quantum associative neural network with nonlinear search algorithm," *International Journal of Theoretical Physics*, vol. 51, no. 3, pp. 705–723, 2012.

[10] D. Rosenbaum, "Binary superposed quantum decision diagrams," *Quantum Information Processing*, vol. 9, no. 4, pp. 463–496, 2010.

[11] J.-P. Tchapet Njafa and S. Nana Engo, "Concise quantum associative memories with nonlinear search algorithm," *Fortschritte der Physik*, vol. 64, no. 2-3, pp. 250–268, 2016.

[12] S. Weinberg, "Precision tests of quantum mechanics," *Physical Review Letters*, vol. 62, no. 5, p. 485, 1989.

[13] D. S. Abrams and S. Lloyd, "Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems," *Phys. Rev. Lett.*, vol. 81, no. 18, pp. 3992–3995, 1998.

[14] D. Ventura and T. Martinez, "Quantum associative memory," *Information Sciences*, vol. 124, no. 1-4, pp. 273–296, 2000.

[15] R. Zhou and Q. Ding, "Quantum pattern recognition with probability of 100%," *International Journal of Theoretical Physics*, vol. 47, no. 5, pp. 1278–1285, 2008.

[16] D. Ventura, "Artificial associative memory using quantum processes," in *Proceedings of the International Conference on Computational Intelligence and Neuroscience*, vol. 2, 1998, pp. 218–221.

[17] A. Leporati, C. Zandron, and G. Mauri, "Simulating the fredkin gate with energy-based p systems." *J. UCS*, vol. 10, no. 5, pp. 600–619, 2004.

[18] A. Leporati and S. Felloni, "Three "quantum" algorithms to solve 3-sat," *Theoretical Computer Science*, vol. 372, no. 2–3, pp. 218 – 241, 2007, membrane Computing.

[19] F. M. de Paula Neto, T. B. Ludermir, W. R. De Oliveira, and A. J. da Silva, "Solving np-complete problems using quantum weightless neuron nodes," in *Intelligent Systems, 2015 Brazilian Conference on*, 2015, pp. 258–263.

[20] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *Journal of statistical physics*, vol. 22, no. 5, pp. 563–591, 1980.

[21] R. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, p. 467, 1982.

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

[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[24] D. S. Abrams and S. Lloyd, "Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems," *Phys. Rev. Lett.*, vol. 81, no. 18, pp. 3992–3995, 1998.

[25] H. Terashima and M. Ueda, "Nonunitary quantum circuit," *eprint arXiv:quant-ph/0304061*, Apr. 2003.

[26] C. Williams and R. Gingrich, "Non-unitary probabilistic quantum computing circuit and method," Aug. 4 2005, uS Patent App. 11/007,792. [Online]. Available: http://www.google.com/patents/US20050167658

[27] M. Ohya and I. Volovich, "Quantum Computing, NP-complete Problems and Chaotic Dynamics," *eprint arXiv:quant-ph/9912100*, Dec. 1999.

[28] A. J. da Silva, T. B. Ludermir, and W. R. de Oliveira, "Quantum perceptron over a field and neural network architecture selection in a quantum computer," *Neural Networks*, vol. 76, pp. 55 – 64, 2016.

[29] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *International Journal of Circuit Theory and Applications*, vol. 39, no. 1, pp. 61–77, 2011.

[30] F. M. de Paula Neto, W. R. de Oliveira, T. B. Ludermir, and A. J. da Silva, "Chaos in a quantum neuron: An open system approach," *Neurocomputing*, pp. –, 2017.

[31] R. M. Gingrich and C. P. Williams, "Non-unitary probabilistic quantum computing," in *Proceedings of the Winter International Synposium on Information and Communication Technologies*, ser. WISICT '04. Trinity College Dublin, 2004, pp. 1–6.

[32] H. Terashima and M. Ueda, "Nonunitary quantum circuit," *International Journal of Quantum Information*, vol. 3, no. 04, pp. 633–647, 2005.

[33] H. Bechmann-Pasquinucci, B. Huttner, and N. Gisin, "Non-linear quantum state transformation of spin-1/2," *Physics Letters A*, vol. 242, pp. 198–204, 1998.

[34] R. O. Duda, P. E. Hart, D. G. Stork *et al.*, *Pattern classification*. Wiley New York, 1973, vol. 2.

APÊNDICE E – QUANTUM PROBABILISTIC ASSOCIATIVE MEMORY ARCHITECTURE

Quantum probabilistic associative memory architecture

Fernando M de Paula Neto[a,*], Adenilton J da Silva[b], Wilson R de Oliveira[b], Teresa B. Ludermir[a]

[a] *Universidade Federal de Pernambuco, Cidade Universitária, Recife, Pernambuco, Brazil*
[b] *Universidade Federal Rural de Pernambuco, Dois Irmãos, Recife, Pernambuco, Brazil*

Abstract

We present a quantum probabilistic associative memory using the inverse of quantum Fourier transform and Grover's algorithm to recover existing or similar patterns in the memory. The content of the memory is created using a generator of a superposition state representing a given set of patterns. We discuss the architecture of the proposed memory including the storing, recovering and processing of similarity tolerance of the input query. The associative memory can extrapolate and recover similar stored patterns. The system is unitary and runs in $O(n)$ steps, where n is the number of qubits of the patterns.

Keywords: Quantum associative memory, Quantum Fourier transform, Quantum search, Grover's algorithm, Quantum computing

1. Introduction

Associative memories are biologically inspired models which store and recover information. A partial or complete input query recovers similar information in the memory. Classically, to be able to restore the information the existing models store m patterns in a given memory needing n neurons, where $m \geq 0.15n$ [1, 2]. In contrast, quantum memories can load 2^n patterns using n neurons [3, 4, 5]. This exponential improvement is possible because the

*Corresponding author
Email address: fmpn2@cin.ufpe.br (Fernando M de Paula Neto)

quantum information processing can be done using a central quantum feature called *superposition*. Superposition allows representing 2^n data in a single n-sized state. Superposition leads to *quantum parallelism*. This parallelism allows processing a function for different values of inputs in a single execution.

Quantum algorithms applied to machine learning is a challenging subject that combines quantum information and machine learning. This new field has been called *quantum machine learning*. Machine learning is a field of Computing Science that extracts information from data to execute noisy or nontrivial process [6]. There are already proposals of quantum machine learning algorithms for some applications, despite the inexistence of a general-purpose quantum computer [7, 8, 9].

Studies involving quantum associative memories have used the notion of search to do the recovery task. The *Grover search algorithm* is a well-known procedure to search a key in an unordered database in $O(\sqrt{N})$ steps where N is the size of the database [10]. Ventura and Martinez [3] proposed a quantum associative memory which runs a modification of the Grover search algorithm with a variable quantum circuit depending on the query state. Ezhov *et al.* extended the work of Ventura and Martinez to include some tolerance of the distance between the query state and the content of the memory not requiring the query to be precisely equal to some stored memory state but closer to it [11]. Trugenberger proposed a probabilistic quantum memory which calculates the Hamming distance of the query and memory states through a Hamiltonian operator and outputs probabilistically the recovered state in the memory [12]. The authors in [13, 4] propose nonlinear quantum associative memories which use nonlinear operators in the recovery procedure.

In this paper, we propose a quantum associative memory that calculates the distance from the query states to the memory states using the quantum Fourier transform. Afterward, we use Grover's algorithm to search the states with distance less than a threshold, increasing the probability of their occurrence. Our main contributions are that our circuit is fixed (independent of the query state); we can consider some tolerance of distance between query and memory

states, which is an essential feature of machine learning classification tools, and the system is linearly bounded by the number of qubits of the patterns to be stored. We set the storage and recovery procedures in a quantum circuit architecture where the parameters of the memory can be varied according to the user preference.

In Section 2 we present concepts of quantum computing including the description of the quantum Fourier transform algorithm and Grover's algorithm. In Section 3 we discuss with more details the quantum associative memories and in Section 4 we present our model of quantum associative memory architecture. The conclusions and final considerations are in Section 5.

2. Quantum computation

2.1. Quantum bits

A *quantum bit* (qubit) is a unit of information in quantum computation - the quantum analog of a binary bit. A qubit is a two-state quantum mechanical system, such as the spin of a spin-$\frac{1}{2}$ particle such as an electron, whose spin can have values $+\hbar/2$ or $-\hbar/2$, where $\hbar$ is the reduced Planck constant. A qubit is represented as a two-dimensional vector in the complex vector space $\mathbb{C}^2$. The two independent (physically distinguishable) quantum states of the system are represented as basis states (or basis vectors) in the canonical (or computational) basis as $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$. As such any qubit can be seen as the linear combination (usually called *superposition*) of the basis vectors as seen in Equation (1),

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are complex numbers. Qubits are normalised, which requires that $|\alpha|^2 + |\beta|^2 = 1$. This notation also means that the qubit has probability $|\alpha|^2$ to be measured as 0 and $|\beta|^2$ to be measured as 1 and α and β are then called *probability amplitudes* or simply *amplitudes*.

The tensor operator $\otimes$ is used to represent quantum systems composed of two or more qubits $|\mathbf{g}\rangle = |ij\rangle = |i\rangle \otimes |j\rangle$. Here we will use the bold font for

the representation of quantum states with more than one qubit. For two qubits $|i\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|j\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, the tensor operator generates the state $|ij\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle$.

For general two vectors $|\mathbf{p}\rangle$ and $|\mathbf{q}\rangle$, respectively living in a n and m-dimensional vector space, their tensor product $|\mathbf{pq}\rangle = |\mathbf{p}\rangle \otimes |\mathbf{q}\rangle$ can be calculated as described in Equation 2 and live in nm-dimensional vector space.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ ... \\ \alpha_n \end{bmatrix} \otimes \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_m \end{bmatrix} = \begin{bmatrix} \alpha_1 \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_m \end{bmatrix} \\ \alpha_2 \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_m \end{bmatrix} \\ ... \\ \alpha_n \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_m \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_2 \\ ... \\ \alpha_1\beta_m \\ \alpha_2\beta_1 \\ \alpha_2\beta_2 \\ ... \\ \alpha_2\beta_m \\ ... \\ \alpha_n\beta_m \end{bmatrix} \tag{2}$$

We can represent the quantum states using integer numbers rather than string of bits inside the $|.\rangle$ notation. For a given n-dimensional quantum state the representation can be $|\boldsymbol{\psi}\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdots + \alpha_n|n\rangle$. The $\langle .|$ notation represents the complex conjugate of the vector $|.\rangle$.

Let Q and R be two vector spaces, their tensor product denoted by $Q \otimes R$, is the vector space generated by the tensor product of all vectors $|a\rangle \otimes |b\rangle$, with $|a\rangle \in A$ and $|b\rangle \in B$.

2.2. *Quantum operators*

Quantum states are modified by quantum operators which change the amplitude values of the qubits. A quantum operator $\mathbf{U}$ acting over a n qubits

system is a unitary complex matrix of order $2^n \times 2^n$. A matrix U is unitary if $U \circ U^\dagger = U^\dagger \circ U = I$, where $U^\dagger$ is the adjoint (conjugate transpose) of U, I is the identity matrix and $\circ$, usually omitted, is the usual matrix product.

Some operators over one qubit are: Identity (**I**), NOT (**X**) and Hadamard (**H**) operators, defined below in Equation (3) and Equation (4) in matrix form and operator form. The combination of these unitary operators forms a quantum circuit.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{matrix} \mathbf{I}|0\rangle = |0\rangle \\ \mathbf{I}|1\rangle = |1\rangle \end{matrix} \qquad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{matrix} \mathbf{X}|0\rangle = |1\rangle \\ \mathbf{X}|1\rangle = |0\rangle \end{matrix} \tag{3}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \begin{matrix} \mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle) \end{matrix} \tag{4}$$

The Identity operator **I** generates the output exactly as the input; **X** operator works as the classic NOT in the computational basis; Hadamard **H** generates a superposition of the basis states with equal probabilities when applied in a computational basis.

In the same way we can combine quantum states, quantum operators can also be combined using tensor product. For two (n_0, m_0)-dimensional matrix U and (n_1, m_1)-dimensional matrix V, their tensor product, $U \otimes V$, is a $(n_0 n_1, m_0 m_1)$-dimensional matrix. We denote as $\mathbf{A}^{\otimes s}$ the s-fold application of **A**.

The **CNOT** is a two qubits operator. It has a control qubit and a target qubit. It works considering the value of the control qubit to apply the **X** operator on the target qubit. If the control qubit is set to $|1\rangle$ the **X** operator is applied to target qubit. The matrix representation for the **CNOT** in the computational basis is shown in Equation 5.

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{matrix} \mathbf{CNOT}|00\rangle = |00\rangle \\ \mathbf{CNOT}|01\rangle = |01\rangle \\ \mathbf{CNOT}|10\rangle = |11\rangle \\ \mathbf{CNOT}|11\rangle = |10\rangle \end{matrix} \tag{5}$$

We can generalise and define a $(n+1)-$ary **CNOT** having n control qubits and requiring all the control qubits to be $|1\rangle$ for applying **X** on the target qubit. We can use the notation n**CNOT** for the case where the **CNOT** has n controllers. To represent which ones are the controllers and target qubits, one can use subscripts, *e.g* $2C\mathbf{CNOT}_{x,y,z}$ where x and y are the controllers' qubits and z is the target qubit.

Considering U as any quantum operator over n qubits, we can generalise the **CNOT** operator idea of control and target qubits further to a controlled-U gate, which has m controlled qubits and applies the U operator to the other n qubits if the controllers are all valued $|1\rangle$.

Any unitary quantum operator can be approximated by a sequence of tensor products of some of the 1-qubit gates seen above and the **CNOT** operator. We say that this set of gates is *universal* for quantum computation in the same vain that the NAND gate is universal for classical Boolean computation.

2.3. Quantum measurement

A general quantum state $|\psi\rangle$ is a superposition of basis states $c_0|\mathbf{0}\rangle + c_1|\mathbf{1}\rangle ... c_{n-1}|\mathbf{n}\rangle$. If one measures this state $|\psi\rangle$ there are probabilities to encounter each one state. It is not possible in the measurement process to check all states present in superposition at once. After a measurement process, the probability to encounter a state $|e\rangle$ is $|\langle\psi||e\rangle|^2$. For example, if a quantum state $|\phi\rangle = \frac{1}{2}|00\rangle + \sqrt{\frac{3}{4}}|11\rangle$ is measured, the probability to encounter the state $|00\rangle$ is $p_{|00\rangle} = \langle\phi||00\rangle = |\frac{1}{2}|^2 = 1/4$.

2.4. Quantum circuit

We can represent quantum operations by quantum circuits. This graphical representation considers the qubits as wires and quantum operators as boxes. The flow of the execution, as in the classical case, is from left to right.

Figure 1 has an example of a quantum circuit composed of a **CNOT**, where a filled circle depicts the control qubit, one **X** operator and one controlled-U operator where here $U = H \otimes H$.

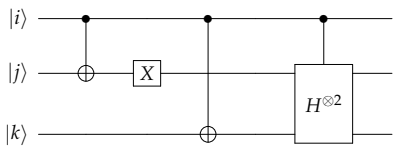


Figure 1: An example of quantum circuit with two **CNOT** operators, one **X** operator and one controlled-U operator where here $U = H \otimes H$.

2.5. Quantum Fourier Transform

The quantum Fourier transform (QFT) algorithm takes into account the phase estimation of a given quantum state. The QFT is a tool widely used in quantum algorithms and essential step in the Shor algorithm, a well-known algorithm in quantum computing that can find prime factors of an integer efficiently [14]. In this Section, we detail the QFT which is used in the recovery procedure of our proposed memory.

The QFT, inspired by the discrete Fourier transform, is the linear operator defined over an orthonormal basis $|0\rangle, \cdots, |N-1\rangle$ of an N-dimensional complex vector space, as:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle \tag{6}$$

In the particular case of n qubits system in the computational basis $|j\rangle = |j_1 j_2 ... j_n\rangle$, where $0 \leq j \leq 2^n - 1$, *i.e.* $N = 2^n$ in Equation 6 above, the QFT can be rewritten as the unitary operator defined as:

$$|j\rangle \rightarrow |j'\rangle = \frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle)...(|0\rangle + e^{2\pi i 0.j_1 j_2 ... j_n}|1\rangle)}{2^{n/2}} \tag{7}$$

The representation $0.j_l j_{l+1} ... jm$ means the binary fraction $\sum_{k=l}^{m} j_k 2^{-k}$. The QFT means that we can represent a given quantum state $|j\rangle$ in a new quantum state $|j'\rangle$ where its phase stores the information of the original quantum

state $|j\rangle$. As the QFT is a unitary operator, its inverse operator can recover the information in the phase of a quantum state, transforming it into a respective quantum state. The definition of the QFT as defined in Equation 7 is called the product representation of the Fourier transform. From this representation it is immediate to generate the quantum circuit shown in Figure 2, where the operator R_x denotes the unitary transformation represented in Equation 8. In Figure 2, it was omitted for simplicity the last step procedure which is the swap operators to reverse the order of the qubits.

$$R_x = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^l} \end{bmatrix} \tag{8}$$

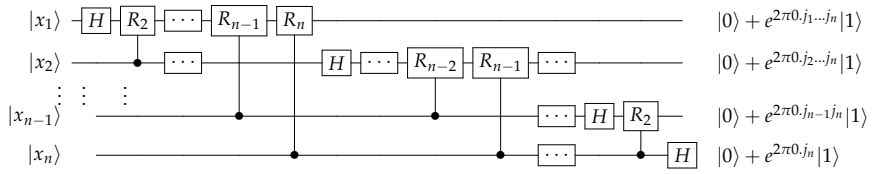


Figure 2: Quantum circuit of the Quantum Fourier Transform

2.6. Grover Search Algorithm

The Grover's algorithm a given fixed basis state amplitude of a given superposed quantum state, increasing the probability of obtaining this basis state after measurement. This procedure can be seen as the task of finding a given key in an unordered database. The N keys are represented as a basis of N-dimensional complex vector space and the database is a general superposed state usually the one with all amplitudes equal to $\frac{1}{\sqrt{N}}$.

Classically, the best algorithm to find a number in an unordered database is a random search algorithm, and it runs $N/2$ steps in average, where N is the size of the database, and $N-1$ steps in the worst case. In quantum com-

puting, representing the database as a state which is a superposition of keys (represented as basis states) one could evaluate the search in parallel in a single operation. The process to amplify the amplitude of some state takes $\sqrt{N}$ steps. In order to identify which state we are looking for, the Grover's algorithm assumes as given an oracle function f as shown in Equation 9. This function f outputs 1 when the input is the expected input x_0 that we are searching and 0 otherwise. More precisely, one needs a quantum oracle U_f which is the quantum realisation of f as $U_f|\mathbf{x}\rangle|y\rangle = |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle$ where $|\mathbf{x}\rangle$ is the key register, $\oplus$ is addition modulus 2, and $|y\rangle$ is a single qubit which when is set to $|0\rangle$ allows one to recover the value of $f(\mathbf{x})$. The operation which amplifies the amplitude of the basis state $|\mathbf{x_0}\rangle$ is detailed in Grover's algorithm in Algorithm 1.

$$f(x) = \begin{cases} 1, & \text{if } x = x_0. \\ 0, & \text{if } x \neq x_0. \end{cases} \tag{9}$$

Step 1. Start with a state $|\mathbf{0}\rangle$

Step 2. Apply $H^{\otimes n}$

Step 3. repeat

> **Step 4.** Apply the phase inversion operation: $U_f(I \otimes H)$;
>
> **Step 5.** Apply the inversion about the mean operation: $-I + 2A$, where I is the 2^n-by-2^n identity matrix and A is the 2^n-by-2^n matrix where any position of that matrix is $1/2^n$;

until $\frac{\pi}{4}\sqrt{2^n}$ *times*;

Step 6. Measure the qubits;

Algorithm 1: Grover's algorithm

The phase inversion operation $U_f(I \otimes H)$ changes in π the phase of the target basis state. The operator $-I + 2A$ execute the inversion about the mean operation that is a step where the target basis amplitude is increased.

Now we present an example of the execution of the Grover's algorithm to find the basis state $|\mathbf{x_0}\rangle = |110\rangle$ in a superposed state with all basis states

having the same amplitude,

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle).$$

In the first iteration of the Grover's algorithm, after the phase inversion operator, the state is

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle - |110\rangle + |111\rangle).$$

After the inversion about mean operation, the state is

$$\frac{1}{2\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + 5|110\rangle + |111\rangle).$$

It is possible to check that after 3 iterations, the amplitude of the desired state $|\mathbf{x_0}\rangle$ will be further amplified.

The original Grover's algorithm amplifies states from a uniform distribution of probabilities. Changes in Grover's algorithm were made to deal with non-uniform distribution of initial states [15, 16]. In this paper, we use the generalised version of Grover's algorithm presented in [15] which deals with arbitrary initial amplitude distribution. The solution presented in [3] is not used in this paper because they used oracles which are dependent on the query and stored states. As a result, for each stored and query states they used a different operator. However, our proposed circuit uses a fixed operator for any input.

3. Quantum associative memories

Traditional computer memories store data sequentially and use addressing to recover information. An associative memory [17] works similar to a biological memory, which are not addressable and where patterns are retrieved from noisy or incomplete content. Some examples of artificial associative networks are the Hopfield network [18] and self-organising maps [19].

In this work, we deal with quantum associative memories that can store information in quantum superposition [20, 12]. A quantum memory has one mechanism to store patterns and other to retrieve information. The main idea

in the storage mechanism is to create an equally distributed superposition of binary patterns. The retrieval mechanism is used to recover one pattern from that superposition (memory), given a noisy or incomplete representation of this pattern as input.

The superposition of binary patterns allows storage with exponential memory capacity concerning the classical ones. A memory of n bits can store 2^n binary patterns. A quantum associative memory was proposed in [3] which uses an adaptation of Grover's algorithm to recovery procedure. This model also offers a pattern storing algorithm [16] which creates a quantum state from a training set for a pattern of length n bits using $2n + 1$ quantum bits. In the retrieval processing, Grover's algorithm is adapted to receive a quantum state which is not all the superposition of equally distributed states. The Grover's algorithm is also adapted to receive a partial (incomplete) quantum state as input. For similar patterns recovery, the authors Ezhov *et al.* extended the work of Ventura and Martinez and included an algorithm that calculates a distance of a query state to each state in the memory [11]. In those papers, there is a probability of success associated with the number of qubits, patterns stored and patterns not stored.

In [21], the quantum storing pattern algorithm uses $2n + 2$ qubits. It executes a unitary retrieval through a Hamiltonian operator which calculates the *Hamming distance* between two binary patterns. The authors argue that the dynamics of the Hamiltonian operator is the generalisation of the Hopfield model and it efficiently finds the exact global minimum of the quantum energy landscape, without the appearance of any spurious memories (*i.e* patterns not desired but possibly included during the training step). However, the output is probabilistic in function of the memory probability distribution. There is also no guarantee that the input is recognised and identified correctly.

In [4], the nonlinear operator $2^n|1\rangle\langle 1|$ is used to recover directly the content in the memory. This nonlinear operator is controversial as regarding it being physical realisable or not. In [5], was proposed to use another nonlinear operator to recover the content of the memory. Although the non-linearity violates

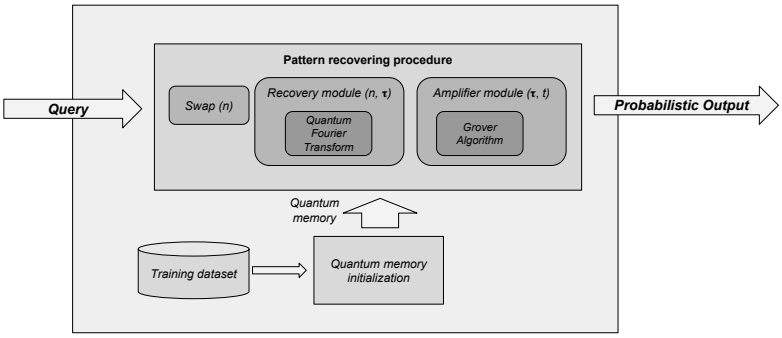


Figure 3: Quantum associative memory architecture.

the quantum mechanics superposition principle, which asserts the linearity of the solutions to the Schrödinger equation, it has been suggested that the quantum system time evolution might be (slightly) nonlinear [22].

The main criticism over quantum associative memories is the memory collapse after processing one input. This problem was first pointed in [23] and raise criticism in [24]. One possible solution to this memory collapse is to use a probabilistic clone machine to preserve the memory [25], but this solution does not solve the decoherence of the quantum state representing the memory. Besides this limitation, quantum associative memories can be used in function optimisation [26] and as a data structure to recover information from a quantum state. This perspective is used in [27], where a quantum associative memory is used as a quantum data structure to evaluate artificial neural networks with an exponential speedup when compared with the best known classical strategies.

4. Quantum associative memory architecture

Our proposed architecture is described in Figure 3. For a given query, the memory output collapses with high probability in a quantum state stored in the

memory whose distance to the query is small enough under some given criteria of tolerance. We explain in subsection 4.1 the module that loads a training dataset and transforms it into a quantum memory. We use the algorithm which generates a superposition of states from a data training set proposed in [21]. Our main contribution in this paper is in the recovering procedure explained in subsection 4.2. Inverse quantum Fourier transform calculates the distance between the query and memory states in a single shot operation (non-iterative). After that, a modified Grover's algorithm is used to amplify the amplitudes of some basis states. The basis states amplified by Grover's algorithm are those which indicates a small distance between the query state and a memory basis state. Examples are presented in subsection 4.4.

4.1. Memory initialisation module

The memory initialisation is an operation that transforms some training set into a superposed quantum state with the amplitudes having the same probability. Given p binary patterns p^i of length n, the memory represented by a quantum state is

$$|m\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} \left|p^i\right\rangle. \tag{10}$$

For our purposes, one can use two initialisation algorithms proposed by Ventura *et al.* [16] and by Trugenberger *et al.* [21]. They give as output a quantum state as described in Equation 10. The complexity of these initialising algorithms is linear in function of the number of patterns in the training set.

In [21], there is a simplified version to generate this entangled state with stored patterns. Three registers are used: a first register p of n qubits in which the patterns p^i to be stored are presented; a utility register u of two qubits prepared in state $|01\rangle$; and another register m of n qubits to hold the memory that is initially prepared in state $|0_1, ..., 0_n\rangle$. The initial quantum state is represented in Equation 11.

$$\left|\psi_0^1\right\rangle = \left|p_1^1, ..., p_n^1; 01; 0_1, ..., 0_n\right\rangle \tag{11}$$

The second qubits of the register u, u_2, indicates, $|0\rangle$ for the stored patterns and $|1\rangle$ for the processing term. For each pattern p^i in the training set to be stored, it is performed the application of the Equation in 12.

$$|\psi_1 i\rangle = \Pi_{j=1}^{n} 2CNOT_{p_j^i u_2 m_j} \left|\psi_0^i\right\rangle \tag{12}$$

The operation in Equation 12 copies the pattern p^i into the memory register of the processing term, identified by $|u_2\rangle = |1\rangle$.

$$\left|\psi_2^i\right\rangle = \Pi_{j=1}^{n} NOT_{m_j} CNOT_{p_j^i m_j} \left|\psi_1^i\right\rangle \tag{13}$$

$$\left|\psi_3^i\right\rangle = nCNOT_{m1\ldots m_n u_1} \left|\psi_2^i\right\rangle \tag{14}$$

In Equation 13, when the contents of the patterns and memory registers are identical, all these qubits are transformed in $|1\rangle$'s. With Equation 14, one changes the first utility qubit u_1 of the processing term to $|1\rangle$, leaving if unchanged for the stored patterns term.

$$\left|\psi_4^i\right\rangle = CS_{u_1 u_2}^{p+1-i} \left|\psi_3^i\right\rangle \tag{15}$$

where CS^i and S^i are defined in Equation 16.

$$\begin{aligned} CS^i &= |0\rangle\langle 0| \otimes 1 + |1\rangle\langle 1| \otimes S^i, \\ S^i &= \begin{pmatrix} \sqrt{\frac{i-1}{i}} & \frac{1}{\sqrt{i}} \\ \frac{-1}{\sqrt{i}} & \sqrt{\frac{i-1}{i}} \end{pmatrix} \end{aligned} \tag{16}$$

In Equation 14, the central operation of the storing algorithm is done. It is separated the new pattern to be stored, already with the correct normalization factor.

$$|\psi_5\rangle = nCNOT_{m1\ldots m_n u_1} \left|\psi_4^i\right\rangle \tag{17}$$

$$\left|\psi_6^i\right\rangle = \Pi_{j=n}^{1} CNOT_{p_j^i m_j} NOT_{m_j} \left|\psi_5^i\right\rangle \tag{18}$$

In Equations 17 and 18, it is applied the inverse of the Equations 13 and 14 which restore the utility qubit u_1 and memory register m to their original values. The three registers after these operation is represented in Equation 19.

$$\left|\psi_6^i\right\rangle = \frac{1}{\sqrt{p}}\sum_{k=1}^{i}\left|p^i;00;p^k\right\rangle + \sqrt{\frac{p-1}{p}}\left|p^i;01;p^i\right\rangle \tag{19}$$

To set the third register m to its initial value $|0_1,...,0_n\rangle$ it is applied as last operation the operation shown in Equation 20. After that, a new pattern is load into the register p and go through the same routine as described. At the end of the process when all patterns in the training set were presented, the m register is the state $|M\rangle$ represented in Equation 10.

$$\left|\psi_7^i\right\rangle = \Pi_{j=n}^{1} 2CNOT_{p_j^i u_2 m_j}\left|\psi_6^i\right\rangle \tag{20}$$

For instance, if we have a training set as $\mathbf{P} = \{0000, 0010, 1010\}$, the quantum memory associated with this training set after the initialising procedure is

$$|m\rangle = \frac{1}{\sqrt{3}}(|0000\rangle + |0010\rangle + |1010\rangle). \tag{21}$$

4.2. Pattern recovering module

Inspired by the quantum perceptron proposed in [28], a module of pattern recovering is built. This recovering process is divided into two steps. In the first step, the distance between the query input and each basis state in the memory is calculated. In this step, the inverse of quantum Fourier algorithm is used. In the second step, the amplification module amplifies the amplitude of basis states which indicates low distance, *i.e* high probability to be the pattern in the memory. In this second step, Grover's algorithm iteratively does this amplification.

The calculus of the distance is done by checking the qubits that are equal between the query and stored qubits in the memory. The algorithm adds in the phase of the query state a Δd angle for each qubit in the query register that is equal to the qubit of the memory register. If there are n qubits in the

query and memory registers, there is the possibility of having 0 to n qubits equal. So, we can represent the number of equal qubits in a quantum register $|\mathbf{c}\rangle$. The rule to define the size of the $|\mathbf{c}\rangle$ register is defined as follows. For a $|\mathbf{c}\rangle$ register with τ qubits, we can check the similarity of the registers of size n, where $n \in [2^{\tau-1}, 2^{\tau} - 1]$. For a given τ size, one defines the value of $\Delta d = \frac{1}{2^{\tau}}$, which is a phase added for each corresponding qubit in the recover execution.

The procedure to check the number of qubits that are equal between $|\mathbf{q}\rangle$ and $|\mathbf{m}\rangle$ is to add in the phase $k\Delta d$ where k is the number of identical qubits between the query and memory states. The circuit presented in Figure 4 does this procedure. The memory state is represented by the register $|\mathbf{m}\rangle$ and the query state by the register $|\mathbf{q}\rangle$.

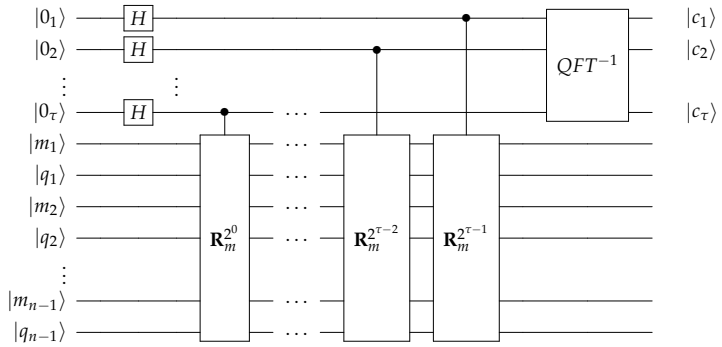


Figure 4: Quantum associative memory recovery module.

For the circuit in Figure 4, the $\mathbf{R}_m$ matrices is composed by the R_m matrix, described in Equation 22, where $\mathbf{R}_m = R_m^{\otimes n}$. The circuit has three registers. The first register has τ zeros. This register is used for load the binary representation of the number of qubits which are equals between memory and the query state. This distance information is saved in the phase of the first register after the application of the successive $\mathbf{R}_m$'s applications. The transformation of the phase information in a quantum state is done by the quantum inverse Fourier

transform operator QFT^{-1}. The second and third registers are the query $|\mathbf{q}\rangle$ and memory $|\mathbf{m}\rangle$ states, where each state has n qubits.

$$R_m = \begin{pmatrix} e^{2\pi i \Delta d} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i \Delta d} \end{pmatrix} \tag{22}$$

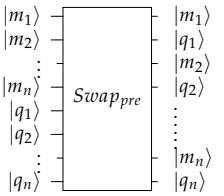


Figure 5: Swapper preprocessing module changes the positions of the memory and query states to input the next module.

The matrix R_m has as input one qubit of the memory and one qubit of the query to be compared. Because of this, we use a swapper module that organises the two registers $|\mathbf{m}\rangle$ and $|\mathbf{q}\rangle$ to intercalate one qubit of each. Figure 5 shows the swapper module.

4.3. Amplifier module

After the comparison of the memory with the query register, we need to check the result. As the memory is associative, it is configured to tolerate some distance different from zero. Hence this module has a function Amp_t which amplifies the amplitudes of the states that have until t qubits different of query input. It implements Grover's algorithm discussed in Section 2.6. The oracle of this procedure U_{oracle} is the operator which changes the phase of the basis states whose values represent the tolerable distance t. In other words, when the size n of the input and query register is $n = 3$, there exist four possibilities

of distance that are $3, 2, 1, 0$. Then the size of τ is enough to be 2, since $2^2 = 4$. The oracle U_{oracle} of the Grover's algorithm will search by the $|\mathbf{3} - \mathbf{0}\rangle = |111\rangle$ basis state if t is set to 0, will search by the $|\mathbf{3} - \mathbf{1}\rangle = |110\rangle$ basis state for $t = 1$, and so on. For $t = 0$, the oracle of the Grover's algorithm U_{oracle} is shown in Equation 23. The representation of the Amp_t operator circuit is shown in Figure 6 following the procedures described in Algorithm 1.

$$U_{oracle} = \sum_{\mathbf{x}=111} -|\mathbf{x}\rangle\langle\mathbf{x}| + \sum_{\mathbf{x}\neq 111} |\mathbf{x}\rangle\langle\mathbf{x}| \tag{23}$$

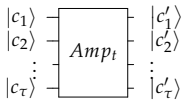


Figure 6: Amplification module circuit Amp_t which amplifies the amplitudes of expected states.

4.4. Examples

4.4.1. Example 1

Consider a given training dataset $\Gamma = \{001, 101, 111\}$. First of all, the memory needs to be initialized. The module of initializing memory generates the memory quantum state $|\mathbf{m}\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |101\rangle + |111\rangle)$. As our memory has patterns with $n = 3$ qubits, τ is chosen to 2. From this configuration, we set $\Delta d = 1/2^2 = 1/4$.

Let consider that the pattern 111 is the query. We feed this state in the pattern recovering procedure as $|\mathbf{q}\rangle = |111\rangle$. The three registers of our memory are $|0_0, 0_1; m_1, ..., m_n; q_1, ..., q_n\rangle = |0_0, ..., 0_\tau\rangle \otimes (\frac{1}{2}(|001\rangle + |101\rangle + |111\rangle)) \otimes |111\rangle$.

After the swap operator, the state will be $|0_0, 0_1; m_1, q_1, m_2, q_2, ..., m_n, q_n\rangle$. The role of the matrices $\mathbf{R_m}$ is to check the memory states qubits that are equal of the query state qubits and add in the phase state the angle Δd when the qubits are equal. If we apply the inversion of the swap operation after the ap-

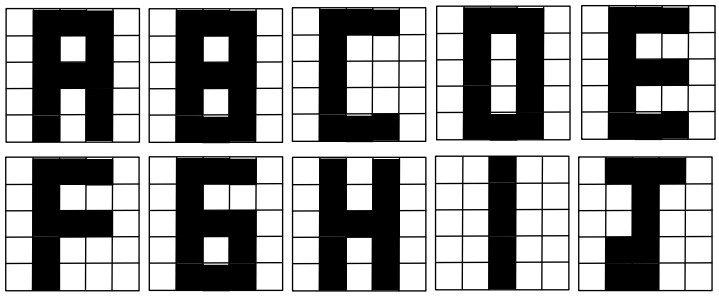


Figure 7: Figures which represent the training dataset to be loaded in the memory.

plication of the $\mathbf{R}_m$ operators, we get the following state $\frac{1}{\sqrt{12}}[(e^{2.\pi.i.\Delta d.2(0)}|00\rangle + e^{2.\pi.i.\Delta d2(1)}|01\rangle + e^{2.\pi.i.\Delta d2(2)}|10\rangle + e^{2.\pi.i.\Delta d2(3)}|11\rangle)|101\rangle|111\rangle + (e^{2\pi i\Delta d1(0)}|00\rangle + e^{2\pi i\Delta d1(1)}|01\rangle + e^{2\pi i\Delta d1(2)}|10\rangle + e^{2\pi i\Delta d1(3)}|11\rangle)|001\rangle|111\rangle + (e^{2\pi i\Delta d3(0)}|00\rangle + e^{2\pi i\Delta d3(1)}|01\rangle + e^{2\pi i\Delta d3(2)}|10\rangle + e^{2\pi i\Delta d3(3)}|11\rangle)|111\rangle|111\rangle]$. Note that the phase in addition to being multiplied by $2\pi i$, so it is also multiplied by the number between 0 to 3 and by decimal value related by its basis state of the first register. The state is in the shape of the right side of Equation 6. For the patterns $|001\rangle$, $|101\rangle$ and $|111\rangle$ the similarities are 2, 1 and 3 respectively in relation to the query state $|111\rangle$. Then the inverse of **QFT** operator will transform this phase in a quantum state as following $|\mathbf{c};\mathbf{m};\mathbf{q}\rangle = \frac{1}{\sqrt{3}}[|01\rangle|001\rangle|111\rangle + |10\rangle|101\rangle|111\rangle + |11\rangle|111\rangle|111\rangle]$. The amplifier module, considering the distance $t = 0$, will amplify the state $|11\rangle$ of the first register. After the application of the amplifier module, the probability of encountering the result $|11\rangle$ is 42.21%. It is possible to measure the result and check the similarity of the memory with the query. If the procedure is repeated 9 times, the probability to check the query state as output increases to 99.28%.

4.4.2. Example 2

In this example, we train our memory with ten letters shown in Figure 7. These figures are converted in binary strings in which 1 denotes black box and

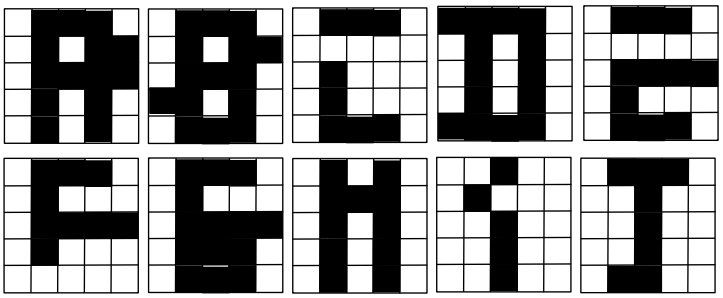


Figure 8: Fault query examples to be recovered in the memory.

0 white box. In Equation 24, one sees the binary strings for each letter where the order of reading is by column.

$$
\begin{aligned}
a &= [0,0,0,0,0,1,1,1,1,1,1,0,1,0,0,1,1,1,1,1,0,0,0,0,0] \\
b &= [0,0,0,0,0,1,1,1,1,1,1,0,1,0,1,1,1,1,1,1,0,0,0,0,0] \\
c &= [0,0,0,0,0,1,1,1,1,1,1,0,0,0,1,1,0,0,0,1,0,0,0,0,0] \\
d &= [0,0,0,0,0,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,0,0,0,0,0] \\
e &= [0,0,0,0,0,1,1,1,1,1,1,0,1,0,1,1,0,1,0,1,0,0,0,0,0] \\
f &= [0,0,0,0,0,1,1,1,1,1,1,0,1,0,0,1,0,1,0,0,0,0,0,0,0] \\
g &= [0,0,0,0,0,1,1,1,1,1,1,0,1,0,1,1,0,1,1,1,0,0,0,0,0] \\
h &= [0,0,0,0,0,1,1,1,1,1,0,0,1,0,0,1,1,1,1,1,0,0,0,0,0] \\
i &= [0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0] \\
j &= [0,0,0,0,0,1,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0]
\end{aligned}
\tag{24}
$$

Our memory transforms the training dataset in a superposition of state $|\mathbf{M}\rangle = \frac{1}{\sqrt{10}}(|000001111110100111110000\rangle + |000001111110101111110000\rangle + |000001111110001100010000\rangle + |000001111110001111110000\rangle + |000001111110101101010000\rangle + |000001111110100101000000\rangle + |000001111110101101110000\rangle + |000001111100100111110000\rangle + |000000000011111000000000\rangle + |000001001111111100000000\rangle)$. For this case,

Table 1: The distances from each query of Figure 8 to the memory states loaded in the memory.

fault query letter q	$d(q,A)$	$d(q,B)$	$d(q,C)$	$d(q,D)$	$d(q,E)$	$d(q,F)$	$d(q,G)$	$d(q,H)$	$d(q,I)$	$d(q,J)$
1	23	22	18	21	20	20	21	22	10	14
2	22	23	19	22	21	19	22	21	11	15
3	19	20	24	21	22	20	21	18	16	20
4	21	22	20	23	20	18	21	20	10	14
5	20	21	21	20	23	21	22	19	15	19
6	20	19	19	18	21	23	20	19	15	17
7	21	22	20	21	22	20	23	20	14	18
8	22	21	19	22	19	19	20	23	11	15
9	14	15	17	14	17	17	16	13	23	19
10	15	16	18	15	18	18	17	14	22	24

we need $\tau = 5$, because at least 25 different distances are allowed when a input query is compared with the memory. Consider that query figures are in Figure 8. In the same way, these query figures are transformed in binary strings and submitted to the memory recovery procedure one at a time. For the first fault query example, $|q\rangle = |000001111110100111111101100\rangle$, the distance from it to each letter A, B, C, D, E, F, G, H, I respectively is 23, 22, 18, 21, 20, 20, 21, 22, 10, 14. The memory procedure can be set to $t = 2$, *i.e.* the tolerance of the distance from query state to memory state is for 2 different boxes. With this configuration, the amplifier module will amplify the amplitude of the basis state which is associated with the letter A. The probability to read the state that represent the figure A is 78.31%. This means that if we repeat the memory execution 3 times, the probability to see the figure A in some iteration is $1 - (1 - 0.3613)^3 \approx 98.97\%$. This probability is not maximum in a single execution because the states are not in a full superposition of states. Only 10 of 2^5 states are being used and, during the Grover's algorithm, states which are not in memory pass to have probability different to zero, appearing in the output what one named as *spurious states*.

One sees in Table 1 that if we set $t = 2$ all the fault query figure can be encountered in the memory with the same probability of 98.97% after three times of execution since the fault query examples have the distance in maximum 2 of their respective memory basis states.

4.5. Uncompleted query

For uncompleted query, the recovery memory process works including in the missing positions all the possible values of qubits. For example, if a state $|001?\rangle$ is searched in the dataset $\Gamma = \{0010, 1001, 1110\}$, it is easy to see that we can transform the query state in $|001\rangle \otimes \mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0010\rangle + |0011\rangle$ and if $t = 0$ the state $|0010\rangle$ will be probably read in the final because it has the minimum distance (zero) to the quantum state in the memory $|0010\rangle$.

4.6. Success probability

The process of recovering the content in the memory involves two parts. The first part calculates the distance of the query input for each memory basis state. The second one amplifies the amplitudes of the states whose distance calculated in the first part is more than one predefined threshold. This amplifying processing is described in [15]. The bound on the probability of measuring a marked state depends only on the standard deviation of the initial amplitude distributions of the marked and unmarked states. Then, our success of probability is almost equal to the presented model in [16]. The difference is that we do not use a variable oracle depending on the input query. As our circuit is fixed, our oracle is a built-in function of the configuration of the memory architecture, not in function of the input query. Then the probability of success is the same as described in [15] that is

$$P(t) = \sum_{i=1}^{r} |k_i(t)|^2 \tag{25}$$

where r is the number of marked states and $k_i(t)$ the amplitudes of the ith marked states in the time t. A bound on this probability is calculated with details in [15] and it is shown in Equation 26 for a given r marked states and

$N - r$ unmarked states and σ_l^2 that is the variance of the amplitudes $\hat{l}$ of the unmarked states is $\frac{1}{N-r}\sum_{i=r+1}^{N}|l_i(t) - \hat{l}(t)^2|$.

$$P_{max} = 1 - (N - r)\sigma_l^2 \tag{26}$$

The time evolution of the amplitudes of marked and unmarked states can be viewed as a first-order linear difference equation with some properties. The essential factor in this analysis is that P_{max} can vary significantly, depending on the statistical properties (average and variance) of the initial amplitude distribution. For more details, we recommend the reader to see [15].

One can see that for the situation were the memory load all the possible states, and we are running the algorithm for the only one basis state, the $\sigma_l^2 = 0$ and $P_{max} = 1$. When there exist more states with zero amplitudes, the probabilities decay. This decay is also verified in [16]. This problem can be bypassed repeating the circuit execution sometimes and checking the result probabilistically. There is a chance to check the query input in the output if exist in the memory $1 - (1 - p)^n$ for the p success probability and n repetition times.

4.7. Circuit complexity

The quantum algorithm to store the information is linear requiring $O(mn)$ steps, where m is the number of patterns and n the length of the patterns. The inverse quantum Fourier transform requires $\frac{\tau(\tau+1)}{2} + 3\frac{\tau}{2}$ gates [29], where τ is the number of qubits where the phase information of the inverse quantum Fourier transform is saved. As τ is less than n, one can consider it takes $O(n)$ steps in this part. In the Grover's algorithm, the iterated procedure is cyclical in $(\pi/4)\sqrt{\tau}$ steps, and thus it takes $O(\sqrt{\tau})$ steps. Finally, considering the biggest complexity, the memory execution has complexity bounded by $O(\sqrt{N} + n) = O(n)$ steps, where $N = 2^\tau = 2^{log_2 n}$.

5. Conclusions

In this paper, we presented a quantum associative memory architecture. Our contributions are in the recovery procedure which produces the distances

from the query state to all memory states using the quantum Fourier transform. In quantum computing, we need to measure the superposed output state to obtain a single value. This measurement destroys the superposition of the states. Then we need to amplify the amplitude of the state whose distance of query state to memory is tolerable. This distance value is the parameter of the Grover oracle which marks the tolerable distance values represented by quantum states. In [16] and [11], the quantum circuit varies depending on the query state. Our proposed model is a parameterised memory depending on the user's configuration but fixed after that configuration. The advantage of keeping the same architecture lies in the difficulty to implement quantum circuits. We need a reprogrammable quantum architecture to generate input dependent circuits [30].

We showed that our model has computational complexity linear in function of the size of the number of qubits of the patterns. Since the storage procedure runs in $O(mn)$ steps, where m is the size of patterns and n the size of qubits in each pattern, the inverse of quantum Fourier runs in $O(n)$ steps and Grover's algorithm runs in $O(\sqrt{2^{log_2 n}})$ steps. The Grover algorithm used in other quantum memory recovery models [3, 11] runs in $O(\sqrt{2^n})$. Since the number of operators in that circuit is proportional with the size of the input, the Grover algorithm whose input size is $log_2 n$, *i.e.* with fewer qubits, is more achievable than those with n to run in a quantum computer available [31].

The memory model can be used to incomplete or fault query. For an incomplete query, the Hadamard operator H is used to generate all the possibility for the missing values. In this way, the process of recovery will identify in the memory the distance from all possibilities to each pattern in the memory and amplify the amplitudes whose distance is short. For fault query, the memory allows to search the patterns with similarity, and that is not entirely equal, using the parameter t to express the tolerance of the distance from the query to each pattern in the memory.

The probability of success of the proposed model is dependent on the pattern number in the memory and the number of states which are marked during

the Grover procedure. If the tolerance of the distance from the query state to memory quantum states is large, more states are marked, and the probability of success is also significant. The bottleneck of the proposed memory is the probabilistic result. All existing methods have probabilistic results, and in our proposed method the tolerable distance is one of the parameters of the memory which can be configured. This problem is bypassed if the procedure of compensation is repeated $\gg n$ times resulting in a probability of success next to 1. Further works aim to maximise this probability of recovering, proposing or adapting quantum search algorithms. Experiments have shown that biologic neurons have chaotic and bifurcation dynamics over their parameters [32, 33, 34]. We have also interest to include a chaotic dynamic [35, 36] during the recovery procedure since that can be evidence to improve the results [37, 38, 39].

Acknowledgement

This work was supported by the Serrapilheira Institute (grant number Serra-1709-22626) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazilian research agency, (Edital Universal, grant number 421849/2016-9 and 442668/2014-7).

References

[1] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences 79 (8) (1982) 2554–2558.

[2] J. W. Bohland, A. A. Minai, Efficient associative memory using small-world architecture, Neurocomputing 38-40 (2001) 489 – 496, computational Neuroscience: Trends in Research 2001.

[3] D. Ventura, T. Martinez, Quantum associative memory, Information Sciences 124 (1-4) (2000) 273–296.

[4] F. M. de Paula Neto, A. J. da Silva, W. R. de Oliveira, T. B. Ludermir, Non-unitary quantum associative memory, in: 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017, pp. 97–102.

[5] R. Zhou, H. Wang, Q. Wu, Y. Shi, Quantum Associative Neural Network with Nonlinear Search Algorithm, International Journal of Theoretical Physics 51 (3) (2012) 705–723.

[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (2017) 195 EP –.

[7] A. J. da Silva, W. R. de Oliveira, T. B. Ludermir, Classical and superposed learning for quantum weightless neural networks, Neurocomputing 75 (1) (2012) 52–60.

[8] M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning, Contemporary Physics 56 (2) (2015) 172–185.

[9] J.-P. T. Njafa, S. N. Engo, Quantum associative memory with linear and non-linear algorithms for the diagnosis of some tropical diseases, Neural Netw. 97 (C) (2018) 1–10.

[10] L. K. Grover, Quantum Mechanics Helps in Searching for a Needle in a Haystack, Phys. Rev. Lett. 79 (2) (1997) 325–328.

[11] A. A. Ezhov, A. V. Nifanova, D. Ventura, Quantum associative memory with distributed queries, Inf. Sci. Inf. Comput. Sci. 128 (3-4) (2000) 271–293.

[12] C. A. Trugenberger, Quantum Pattern Recognition, Quantum Information Processing 1 (6) (2002) 471–493.

[13] R. Zhou, H. Wang, Q. Wu, Y. Shi, Quantum associative neural network with nonlinear search algorithm, International Journal of Theoretical Physics 51 (3) (2012) 705–723.

[14] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (5) (1997) 1484–1509.

[15] D. Biron, O. Biham, E. Biham, M. Grassl, D. A. Lidar, Generalized grover search algorithm for arbitrary initial amplitude distribution, in: C. P. Williams (Ed.), Quantum Computing and Quantum Communications, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 140–147.

[16] D. Ventura, T. Martinez, Initializing the amplitude distribution of a quantum state, Foundations of Physics Letters 12 (6) (1999) 547–559.

[17] T. Kohonen, Self-organization and associative memory, Vol. 8, Springer Science & Business Media, 2012.

[18] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the national academy of sciences 79 (8) (1982) 2554–2558.

[19] T. Kohonen, Self-organized formation of topologically correct feature maps, Biological cybernetics 43 (1) (1982) 59–69.

[20] D. Ventura, T. Martinez, Quantum associative memory, Information Sciences 124 (1–4) (2000) 273 – 296.

[21] C. A. Trugenberger, Probabilistic Quantum Memories, Phys. Rev. Lett. 87 (6) (2001) 67901.

[22] D. S. Abrams, S. Lloyd, Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems, Phys. Rev. Lett. 81 (18) (1998) 3992–3995.

[23] T. Brun, H. Klauck, A. Nayak, M. Rötteler, C. Zalka, Comment on "probabilistic quantum memories", Phys. Rev. Lett. 91 (2003) 209801.

[24] V. Dunjko, H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: a review of recent progress, Reports on Progress in Physics.

[25] C. A. Trugenberger, Trugenberger replies, Physical Review Letters 91 (20) (2003) 209802.

[26] C. A. Trugenberger, Quantum optimization for combinatorial searches, New Journal of Physics 4 (1) (2002) 26.

[27] A. J. da Silva, R. L. F. de Oliveira, Neural networks architecture evaluation in a quantum computer, in: 2017 Brazilian Conference on Intelligent Systems (BRACIS), 2017, pp. 163–168.

[28] M. Schuld, I. Sinayskiy, F. Petruccione, Simulating a perceptron on a quantum computer, Physics Letters A 379 (7) (2015) 660 – 663.

[29] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.

[30] E. H. Knill, Building quantum computers, IEEE Information Theory Society News Letter 58 (IEEE Information Theory Society News Letter).

[31] J. Preskill, Quantum computing in the nisq era and beyond, arXiv preprint arXiv:1801.00862.

[32] H. Korn, P. Faure, Is there chaos in the brain? ii. experimental evidence and related models, Comptes Rendus Biologies 326 (9) (2003) 787 – 840.

[33] W. J. FREEMAN, Tutorial on neurobiology: From single neurons to brain chaos, International Journal of Bifurcation and Chaos 02 (03) (1992) 451–482.

[34] G. Matsumoto, K. Aihara, Y. Hanyu, N. Takahashi, S. Yoshizawa, J. ichi Nagumo, Chaos and phase locking in normal squid axons, Physics Letters A 123 (4) (1987) 162 – 166.

[35] F. M. de Paula Neto, W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, Chaos in Quantum Weightless Neuron Node Dynamics, Neurocomputing 183 (2015) 23–38.

[36] F. M. de Paula Neto, W. R. de Oliveira, T. B. Ludermir, A. J. da Silva, Chaos in a quantum neuron: An open system approach, Neurocomputing 246 (2017) 3 – 11.

[37] G. He, L. Chen, K. Aihara, Associative memory with a controlled chaotic neural network, Neurocomputing 71 (13) (2008) 2794 – 2805, artificial Neural Networks (ICANN 2006) / Engineering of Intelligent Systems (ICEIS 2006).

[38] D. Yang, G. Li, G. Cheng, On the efficiency of chaos optimization algorithms for global optimization, Chaos, Solitons, Fractals 34 (4) (2007) 1366 – 1375.

[39] T. Quang, D. Khoa, M. Nakagawa, Neural network learning based on chaos.