Pós-Graduação em Ciência da Computação

ARTIFICIAL NEURAL NETWORK ARCHITECTURE SELECTION IN A QUANTUM COMPUTER

Por

Adenilton José da Silva

Tese de Doutorado

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

RECIFE
2015

Adenilton José da Silva

Artificial Neural Network Architecture Selection in a Quantum Computer

Esta tese foi submetida 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.

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

Recife
2015

Catalogação na fonte
Bibliotecária Joana D'Arc Leão Salvador CRB4-532

S586a Silva, Adenilton José da.
Artificial neural network architecture selection in a quantum computer / Adenilton José da Silva. – Recife: O Autor, 2015.
77 f.: fig., tab.

Orientadora: Teresa Bernarda Ludermir.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIN, Ciência da Computação, 2015.
Inclui referências.

1. Inteligência artificial. 2. Redes neurais (Computação).
3. Computação quântica. I. Ludermir, Teresa Bernarda (Orientadora). II. Titulo.

006.3 CDD (22. ed.) UFPE-MEI 2015-091

Tese de Doutorado apresentada por **Adenilton José da Silva** à Pós Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Artificial Neural Network Architecture Selection in a Quantum Computer**" orientada pela **Profa. Teresa Bernarda Ludermir** e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Borko Stosic
Departamento de Estatística e Informática / UFRPE

_____

Prof. André Carlos Ponce de Leon Ferreira de Carvalho
Instituto de Ciências Matemáticas e de Computação / USP

_____

Prof. Antônio Murilo Santos Macêdo
Departamento de Física / UFPE

_____

Profa. Marley Maria Bernardes Rebuzzi Vellasco
Departamento de Engenharia Elétrica / PUC-RJ

_____

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

Visto e permitida a impressão.
Recife, 26 de junho de 2015.

_____

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

I dedicate this thesis to all my family, friends and teachers who gave me the necessary support to get here.

Acknowledgements

Não só durante a elaboração desta tese e sim durante os últimos 9 anos tive minha vida acadêmica acompanhada pelos meus orientadores Wilson de Oliveira e Teresa Ludermir. Começo este agradecimento pelos meus orientadores, pois sem eles este trabalho não teria sido possível.

No início de 2007 meu irmão Adilton Silva me apresentou o professor Wilson de Oliveira. Na época eu não poderia imaginar que aquele momento iria definir grande parte da minha vida. O professor Wilson demonstrou características marcantes nesta primeira impressão, como por exemplo, sua excelente criatividade produtiva, paixão pela pesquisa e a humildade de ouvir com toda atenção as dúvidas e sugestões de seus alunos. Hoje percebo que Wilson como meu orientador teve e tem um grande papel no meu interesse pela pesquisa e também tem grande influência em todos os campos da minha vida profissional e pessoal.

Também em 2007 conheci a professora Teresa Ludermir que é minha orientadora desde a iniciação científica. Teresa é uma pesquisadora notável, com alta produção científica, atenciosa e sua dedicação a orientação deste e outros trabalhos teve um papel fundamental na minha formação como pesquisador. A experiência da professora Teresa foi fundamental para o desenvolvimento desta tese, ela soube me dar a liberdade de pensamento necessária para a construção desta tese e nos momentos necessários também soube me alertar para dar um rumo correto ao desenvolvimento desta tese.

Wilson e Teresa, meus sinceros agradecimentos por terem me acompanhado na minha jornada até o momento e pela dedicação e atenção que me deram nesta última década.

Durante os últimos 5 anos diversas pessoas contribuíram de forma direta ou indireta na construção desta tese. Preciso destacar nominalmente Nicole Luana Mineu e Thiago Pessoa Ferreira de Lima com quem tive a oportunidade de conversar em épocas distintas sobre métodos para seleção de arquitetura em redes neurais clássicas; e Fernando Maciano de Paula Neto com quem tive a oportunidade de conversar sobre redes neurais quânticas estas conversas resultaram em alguns trabalhos não diretamente relacionados a esta tese e também me inspiraram a buscar um algoritmo quântico para a seleção de arquiteturas de redes neurais.

Agradeço também a todos os professores do Centro de Informática da Universidade Federal de Pernambuco e do departamento de Matemática da Universidade Federal Rural de Pernambuco pelos concelhos, orientações e pelo estímulo a carreira acadêmica. Sem o auxílio destes profissionais o caminho que levou ao início deste trabalho não seria possível.

Agradeço a todos os funcionários da Universidade Federal Rural de Pernambuco e da Universidade Federal de Pernambuco que sempre me atenderam e resolveram todos os meus pedidos com simpatia e graciosidade e também por manterem um clima propício a pesquisa e colaboração. Agradeço também a todos os demais alunos do Centro de Informática da UFPE com quem dividi os espaços comuns e laboratórios de pesquisa por todos os momentos de trabalho e de descontração. Entre estes alunos não poderia deixar de citar nominalmente Valmir Macário Filho, Luciano Demétrio Santos Pacífico e Péricles Barbosa Cunha de Miranda agora são meus colegas de trabalho no DEInfo - UFRPE.

Agradeço a todos os meus colegas de trabalho da Universidade Federal Rural de Pernambuco. Minha história nesta Universidade é maior do que o espaço deste agradecimento. Grande parte desta tese foi escrita no departamento de Estatística e Informática desta universidade.

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes).

Finalmente, agradeço a toda minha família por todo o incentivo e apoio que foram essenciais para que este trabalho fosse possível. Agradeço aos meus pais que sempre me permitiram escolher meu caminho e me apoiaram incondicionalmente em todas as minhas decisões. Ao meu irmão Adilton Silva por ter me incentivado a seguir uma carreira acadêmica. Um agradecimento especial a minha esposa e ao meu filho pelo apoio emocional e compreensão em todos os momentos.

"We can only see a short distance ahead,
but we can see plenty there that needs to be done"
– Alan Turing

Abstract

Miniaturisation of computers components is taking us from classical to quantum physics domain. Further reduction in computer components size eventually will lead to the development of computer systems whose components will be on such a small scale that quantum physics intrinsic properties must be taken into account.

The expression quantum computation and a first formal model of a quantum computer were first employed in the eighties. With the discovery of a quantum algorithm for factoring exponentially faster than any known classical algorithm in 1997, quantum computing began to attract industry investments for the development of a quantum computer and the design of novel quantum algorithms. For instance, the development of learning algorithms for neural networks.

Some artificial neural networks models can simulate an universal Turing machine, and together with learning capabilities have numerous applications in real life problems. One limitation of artificial neural networks is the lack of an *efficient* algorithm to determine its optimal architecture. The main objective of this work is to verify whether we can obtain some advantage with the use of quantum computation techniques in a neural network learning and architecture selection procedure.

We propose a quantum neural network, named quantum perceptron over a field (QPF). QPF is a direct generalisation of a classical perceptron which addresses some drawbacks found in previous models for quantum perceptrons. We also present a learning algorithm named Superposition based Architecture Learning algorithm (SAL) that optimises the neural network weights and architectures. SAL searches for the best architecture in a finite set of neural network architectures and neural networks parameters in linear time over the number of examples in the training set. SAL is the first quantum learning algorithm to determine neural network architectures in linear time. This speedup is obtained by the use of quantum parallelism and a non linear quantum operator.

Key-words: Quantum neural networks. Artificial neural networks. Quantum computation. Architecture selection.

Resumo

A miniaturização dos componentes dos computadores está nos levando dos domínios da física clássica aos domínios da física quântica. Futuras reduções nos componentes dos computadores eventualmente levará ao desenvolvimento de computadores cujos componentes estarão em uma escala em que efeitos intrínsecos da física quântica deverão ser considerados.

O termo computação quântica e um primeiro modelo formal de computação quântica foram definidos na década de 80. Com a descoberta no ano de 1997 de um algoritmo quântico para fatoração exponencialmente mais rápido do que qualquer algoritmo clássico conhecido a computação quântica passou a atrair investimentos de diversas empresas para a construção de um computador quântico e para o desenvolvimento de algoritmos quânticos. Por exemplo, o desenvolvimento de algoritmos de aprendizado para redes neurais.

Alguns modelos de Redes Neurais Artificiais podem ser utilizados para simular uma máquina de Turing universal. Devido a sua capacidade de aprendizado, existem aplicações de redes neurais artificiais nas mais diversas áreas do conhecimento. Uma das limitações das redes neurais artificiais é a inexistência de um algoritmo com custo polinomial para determinar a melhor arquitetura de uma rede neural. Este trabalho tem como objetivo principal verificar se é possível obter alguma vantagem no uso da computação quântica no processo de seleção de arquiteturas de uma rede neural.

Um modelo de rede neural quântica denominado perceptron quântico sobre um corpo foi proposto. O perceptron quântico sobre um corpo é uma generalização direta de um perceptron clássico que resolve algumas das limitações em modelos de redes neurais quânticas previamente propostos. Um algoritmo de aprendizado denominado algoritmo de aprendizado de arquitetura baseado no princípio da superposição que otimiza pesos e arquitetura de uma rede neural simultaneamente é apresentado. O algoritmo proposto possui custo linear e determina a melhor arquitetura em um conjunto finito de arquiteturas e os parâmetros da rede neural. O algoritmo de aprendizado proposto é o primeiro algoritmo quântico para determinar a arquitetura de uma rede neural com custo linear. O custo linear é obtido pelo uso do paralelismo quântico e de um operador quântico não linear.

Palavras-chave: Redes neurais quânticas. Computação quântica. Redes neurais artificiais. Seleção de arquitetura.

List of abbreviations and acronyms

ANN	Artificial Neural Network
qANN	Altaisky quantum Neural Network
QNN	Quantum Neural Network
PLN	Probabilistic Logic Node
PA	Probabilistic Automata
WNN	Weightless Neural Network
qMPN	Quantum M-P Neural network
NNQA	Neural Network with Quantum Architecture
qRAM	quantum RAM neuron
qPLN	quantum Probabilistic Logic Node
QPF	Quantum Perceptron over a Field
SAL	Superposition based Architecture Learning algorithm
RAM	Random Access memory
LMS	Least Mean Square
qWNN	quantum Weightless Neural Network
SLA	Superposition-based Learning Algorithm
MPLN	Multi-valued Probabilistic Logic Node
GSN	Goal Seeking Neuron
pRAM	probabilistic RAM node
qGSN	quantum Goal Seeking Neuron
VG-RAM	Virtual Generalizing RAM

Contents

1 Introduction

This thesis deals with learning and architecture selection of artificial neural networks using a quantum computer. The focus is on quantum neural networks and learning algorithms that are mathematically described to work on a quantum computer. The main result is about neural network architecture selection. It is shown that with quantum computing and with a nonlinear quantum operator it is possible to select the best neural network architecture for a given dataset in linear time.

The theoretical possibility of quantum computing was initiated with Benioff (1980) and Feynman (1982) and the formalization of the first quantum computing model was proposed by Deutsch (1985). In (BENIOFF, 1980) it is shown through the simulation of a Turing Machine that it is possible to use quantum mechanics to model a decision procedure. In (FEYNMAN, 1982) it is pointed that a quantum system cannot be efficiently simulated on a classical computer; on the other hand, a quantum system can be simulated on a quantum computer with exponential gain when compared with classical computing. Feynman concluded that a quantum computer should have a greater computational power than a classical computer. Deutsch (1985) proposes a quantum Turing machine and important concepts as universal quantum Turing machine and quantum parallelism are introduced.

The concept of quantum parallelism is one of the main characteristics of quantum computing. Quantum parallelism allows the application of a function to several (possibly all) values with the cost of only a single function application. Quantum parallelism has been used to develop quantum algorithms that can solve some problems more efficiently than any known classical algorithm. For instance, a factoring algorithm with polynomial time (SHOR, 1997), a search algorithm for unordered data with quadratic gain in relation to the best classical algorithm (GROVER, 1997) and an algorithm to simultaneously present all examples in one dataset for a machine learning algorithm (VENTURA; MARTINEZ, 1999). Quantum parallelism cannot be directly used, because it is not possible to directly see the results obtained after the function evaluation. Measurement in quantum computing is probabilistic and disturbs the system.

An explosive development of quantum technologies and quantum information fields occurred in the nineties (GEORGESCU, 2014). Quantum computers are not yet a reality, but several research groups are working on the development of quantum computers. Advances in the development of a quantum computer have been done. For instance, in (MONZ et al., 2011) it is reported the development of a system with 14 quantum bits (qubits) and in (BIAN et al., 2013) it is reported the development of a system with 84

qubits. Actually, some quantum operations can be performed, for instance, long distance quantum communication (DUAN et al., 2001) that promises secure transmission and transfer of quantum states; in (JENNEWEIN et al., 2000) a quantum device to generate random numbers is presented (current theory states that generation of true random numbers can be performed only with quantum mechanics).

A quantum computer useful for artificial neural networks simulation needs of a quantum memory with capacity to manipulate hundreds of qubits. Such technology does not exist and all concepts analysed in this work cannot yet be experimentally verified.

The remainder of this Chapter is organized in the following way. Section 1.1 presents the motivation for quantum neural computing. Section 1.2 presents the objectives of this work. Section 1.3 presents some models related to quantum neural networks that are out of the scope of this work. Section 1.4 presents the contributions of this work. Section 1.5 presents the bibliographical production. Section 1.6 describes the organization of this work. And Section 1.7 presents the Thesis conclusion.

1.1 Motivation

1.1.1 From neural to quantum computation

Artificial neural networks (ANN) (HAYKIN, 1999) are a universal model of computation (CABESSA; SIEGELMANN, 2014) with learning capabilities and have several applications in real life problems. There are problems without known algorithmic solution, but an ANN can induce a map input/output of the problem. For instance, in the identification of factors related to common mental disorders (LUDERMIR; DE OLIVEIRA, 2013), stock market forecasting (TICKNOR, 2013; KOURENTZES; BARROW; CRONE, 2014), in the analysis of service quality in public transportation (GARRIDO; DE OÑA; DE OÑA, 2014), air quality analysis (MATTOS NETO et al., 2014), in the solution of combinatorial optimisation problems (ZHANG et al., 2014), pattern recognition (GONZÁLEZ et al., 2014) and in several other applications (MACARIO et al., 2013; PACIFICO; LUDERMIR, 2013; STAFFA et al., 2014; LIMA; LUDERMIR, 2013).

ANN have some problems as the lack of an algorithm to determine optimal architectures (YAMAZAKI; LUDERMIR, 2003), low memory capacity when used as associative memory and high cost learning algorithms (BEIGY; MEYBODI, 2001). Neural networks application depends on the choice of the neural network architecture. An empirical, costly and repetitive process to determine the neural network architecture normally is performed by an expert. Several heuristics have been proposed to automatize the choice of neural network architecture. Some optimization strategies used in heuristics to choose neural

networks architectures are evolutionary algorithms (DA SILVA; MINEU; LUDERMIR, 2010; ALMEIDA; LUDERMIR, 2010), meta-learning (ABRAHAM, 2004; KORDÍK et al., 2010) and genetic programming (KOZA; RICE, 1991).

A matrix of weights is a common representation of a neural network in software. Output of a neural network layer is the product of the matrix of weights with an input vector followed by the application of a nonlinear activation function. Quantum operators are also represented by matrices and quantum bits are represented by vectors. And attempts to represent a neural network as a quantum operator have been proposed in several works (ZHOU; DING, 2007; ALTAISKY, 2001; PANELLA; MARTINELLI, 2011).

Another supposition that led the study of neural networks to the quantum domain is that classical models of computation may not be used to simulate biological model of intelligence and quantum models must be used to simulate consciousness (PENROSE, 1999). This is a controversial statement and in (TEGMARK, 2000) it is stated that the brain is probably not a quantum computer, because neuron state superposition involves millions of ions and the decoherence time of this superposition will be smaller than the neuron action. An answer to this question is an open problem besides some works shown the improbability of necessity to use quantum computing to simulate cognitive processes (KOCH; HEPP, 2006). Independent of which affirmation is true, notions of Quantum Neural Networks have been put forward since the nineties (KAK, 1995), but a precise definition of what is a quantum neural network that integrates neural computation and quantum computation is a non-trivial open problem (SCHULD; SINAYSKIY; PETRUCCIONE, 2014b). To date, the majority of proposed models in the literature are really just quantum inspired in the sense that despite using quantum representation of data and quantum operators, in a way or another some quantum principles are violated usually during training.

1.1.2 From quantum to neural computation

Quantum computing is a new subject and a multidisciplinary field (mathematics, physics, computer science and engineering). To develop a quantum algorithm is not an easy task (we desire quantum algorithms that overcome the classical algorithms). Two possible reasons for the lack of new quantum algorithms are that computer scientists do not know quantum computation or there are only few interesting quantum algorithms (SHOR, 2003). On the other hand, several companies work to develop quantum computers and quantum algorithms (MORET-BONILLO, 2014).

One attempt to expand quantum computing application area is to develop an intelligent system based on quantum computing principles (MORET-BONILLO, 2014). These intelligent systems will allow the application of quantum computing in a greater range of problems (FARHI; GUTMANN, 1998; MALOSSINI; CALARCO, 2008). Several companies and

research institutions have put effort in develop quantum machine learning techniques, for instance, the quantum artificial intelligence lab (NASA, 2013; NEVEN, 2013) launched by Google, NASA and Universities Space Research Association; the Microsoft research group in quantum artificial intelligence (MICROSOFT, 2014); and several researchers around the world (BEHRMAN et al., 2000; TRUGENBERGER, 2002; RICKS; VENTURA, 2004; PANELLA; MARTINELLI, 2011; DA SILVA; DE OLIVEIRA; LUDERMIR, 2012; ALTAISKY; KAPUTKINA; KRYLOV, 2014).

Quantum computing has been used in the development of new machine learning techniques as quantum decision trees (FARHI; GUTMANN, 1998), artificial neural networks (NARAYANAN; MENNEER, 2000; PANELLA; MARTINELLI, 2011; DA SILVA; DE OLIVEIRA; LUDERMIR, 2012), associative memory (VENTURA; MARTINEZ, 2000; TRUGENBERGER, 2001), nearest-neighbour learning algorithms (WIEBE; KAPOOR; SVORE, 2015), quantum learning algorithm for support vector machines (REBENTROST; MOHSENI; LLOYD, 2014) and inspired the development of novel evolutionary algorithms for continuous optimization problems (HSU, 2013; DUAN; XU; XING, 2010). There is an increasing interest in quantum machine learning and more specifically in the quantum neural network area (SCHULD; SINAYSKIY; PETRUCCIONE, 2014b).

1.1.3 Quantum neural computation

Research in quantum neural computing is unrelated, as stated in (SCHULD; SINAYSKIY; PETRUCCIONE, 2014b):

> "QNN research remains an exotic conglomeration of different ideas under the umbrella of quantum information".

Currently there is no consensus of what are the components of a quantum neural network. Several models of quantum neural networks have been proposed and they present different conceptual models. In some models a quantum neural network is described as a physical device (NARAYANAN; MENNEER, 2000); as a model only inspired in quantum computing (KOUDA et al., 2005); or as a mathematical model that explores quantum computing principles (ZHOU; DING, 2007; PANELLA; MARTINELLI, 2011; DA SILVA; DE OLIVEIRA; LUDERMIR, 2012; SCHULD; SINAYSKIY; PETRUCCIONE, 2014a). We follow the last approach and assume that our quantum neural network model would be implemented in a quantum computer that follows the quantum principles as *e.g.* described in (NIELSEN; CHUANG, 2000). We assume that our model is implemented in the quantum circuit model of Quantum Computing (NIELSEN; CHUANG, 2000).

Some advantages of quantum neural models over the classical models are the exponential gain in memory capacity (TRUGENBERGER, 2002), quantum neurons can solve

non-linearly separable problems (ZHOU; DING, 2007), and a nonlinear quantum learning algorithm with linear time over the number of examples in the data set is presented in (PANELLA; MARTINELLI, 2011). However, these quantum neural models cannot be viewed as a direct generalization of a classical neural network.

1.2 Objectives

The use of artificial neural networks to solve a problem requires considerable time for choosing parameters and neural network architecture (ALMEIDA; LUDERMIR, 2010). The architecture design is extremely important in neural network applications because a neural network with a simple architecture may not be capable to perform the task. On the other hand, a complex architecture can overfit the training data (YAMAZAKI; LUDERMIR, 2003). The definition of an algorithm to determine (in a finite set of architectures) the best neural network architecture (minimal architecture for a given learning task that can learn the training dataset) efficiently is an open problem.

The objective of this work is to verify whether we can obtain some advantage with the use of quantum computation techniques in a neural network learning and architecture selection procedure. More specifically this work shows that with the supposition of non-linear quantum computing (PANELLA; MARTINELLI, 2011; PANELLA; MARTINELLI, 2009; ABRAMS; LLOYD, 1998) we can determine an optimal neural network architecture that can learn the training data in linear time with relation to the number of examples in the training set. To accomplish this objective, previously proposed quantum neural networks models were analysed to verify the actual development of quantum neural networks field and if the proposed quantum neural networks models respect quantum and neural computing principles. It was necessary to propose a quantum neural network that respect the principles of quantum computation, neural computing and generalizes the classical perceptron. The proposed neuron works as a classical perceptron when the weights are in the computational basis, but as quantum perceptron when the weights are in superposition.

We propose a quantum nonlinear neural network learning algorithm which uses a non-linear quantum operator (PANELLA; MARTINELLI, 2011; ABRAMS; LLOYD, 1998) to perform a global search in the space of weights and architecture of a neural network. The proposed learning algorithm is the first quantum algorithm performing this kind of optimization in linear time and presents a framework to develop linear quantum learning algorithms to find near optimal neural network architectures.

Specific objectives of this work are

1. To develop a quantum neural network based on weightless neural network and its learning algorithm.

2. To investigate quantum neural network based on the quantum multi-layer perceptron. And if necessary, define a quantum perceptron. This quantum neural network must have the following characteristics: direct implementation in quantum circuits; capacity to simulate the classical models; and allow the use of a quantum learning algorithm.

3. To develop a learning algorithm for quantum neural networks. The learning algorithm must be capable to select neural network free parameters and architecture.

1.3 Out of scope

Quantum computing and neural networks are multidisciplinary research fields. In this way, the quantum neural computing research is also multidisciplinary and concepts from physics, mathematics and computer science are used. Probably because of this multidisciplinary characteristic there are completely different concepts named quantum neural networks. In this Section, we point some models that are out of the scope of this thesis.

1.3.1 Quantum inspired neural networks

Neural networks whose definition is based on quantum computation, but that works in a classical computer as in (KOUDA et al., 2005; ZHOU; QIN; JIANG, 2006; LI et al., 2013) are named in this work as Quantum Inspired Neural Networks. Quantum inspired neural networks are not real quantum models. Quantum inspired models are classical neural networks that are inspired in quantum computing exactly as there are combinatorial optimization methods inspired in ant colony or bird swarm.

In (KOUDA et al., 2005) a complex neural network named qubit neural network whose neurons acts in the phase of the input values is proposed. The qubit neural network has its functionality based on quantum operation, but it is a classical model and can be efficiently simulated in a classical computer.

Another quantum inspired models is defined in (ZHOU et al., 1999) where the activation function is a linear combination of sigmoid functions. This linear combination of activation functions is inspired in the concept of quantum superposition, but these models can be efficiently simulated by a classical computer.

Quantum inspiration can bring useful new ideas and techniques for neural network models and learning algorithms design. However quantum inspired neural networks are out of the scope of this work.

1.3.2 Physical device quantum neural networks

Devices that implement a quantum neural network are proposed in (NARAYANAN; MENNEER, 2000; BEHRMAN et al., 2000). In this work, these models are named physical device quantum neural network. The main problem of this kind of proposal is the hardware dependence. Scalable quantum computers are not yet a reality and when someone build a quantum computer we do not know which hardware will be used.

In (NARAYANAN; MENNEER, 2000) a quantum neural network is represented by the architecture of a double slit experiment where input examples are represented by photons, neurons are represented by slits, weights are represented by waves and screen represents output neurons. In (BEHRMAN et al., 2000) a quantum neural network is represented by a quantum dot molecule evolving in real time. Neurons are represented by states of molecules, weights are the number of excitations that are optically controlled, inputs are the initial state of the quantum dot molecules and outputs are the final state of the dot molecules.

Physical device quantum neural networks are real quantum models. This kind of quantum neural networks needs of specific hardware and is out of the scope of this work.

1.3.3 Quantum inspired algorithms

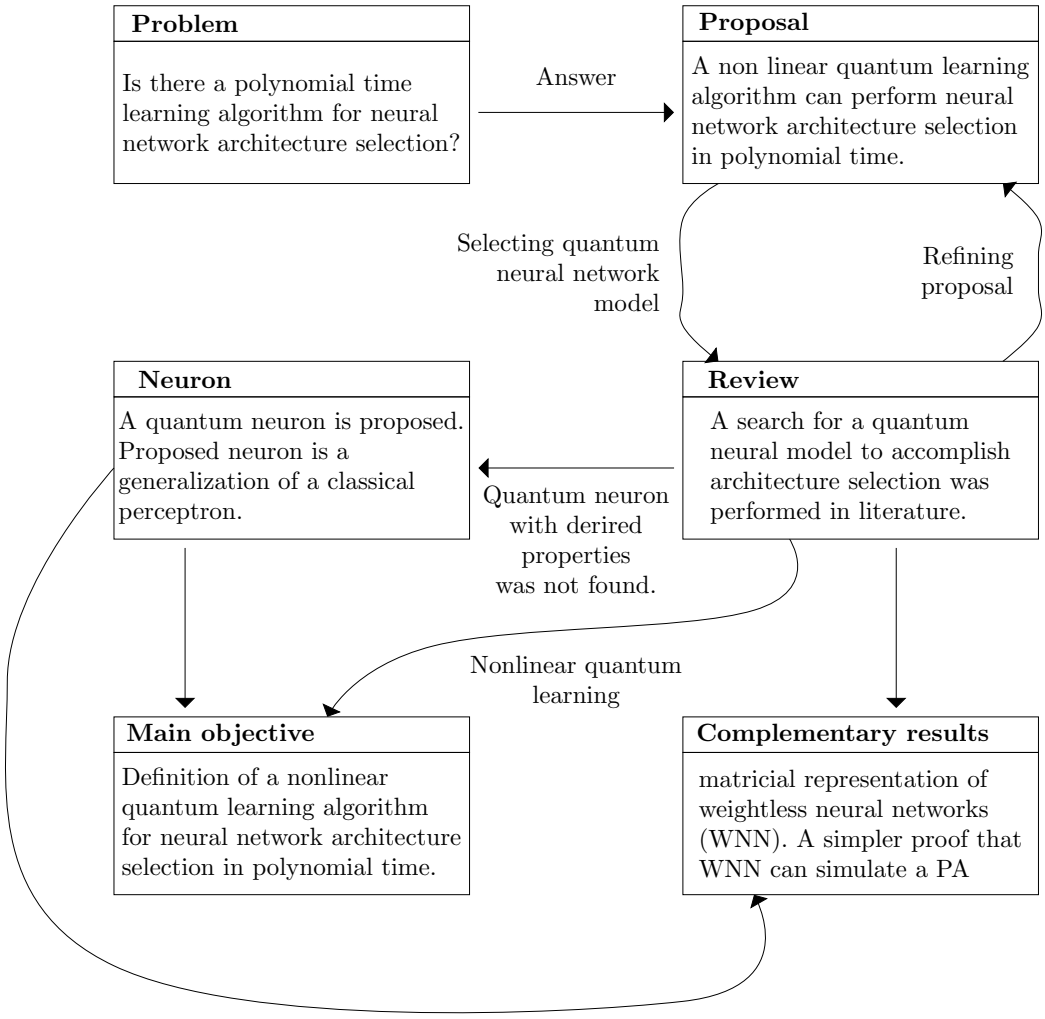
In this work algorithms whose development uses ideas from quantum computing, but run in a classical computer are named quantum inspired algorithms. For instance, there are several quantum inspired evolutionary algorithm proposed in literature (HAN; KIM, 2002; TIRUMALA; CHEN; PANG, 2014; STRACHAN et al., 2014; DA CRUZ; VELLASCO; PACHECO, 2006). In (STRACHAN et al., 2014) a quantum inspired evolutionary algorithm is proposed. This kind of algorithm uses quantum inspiration to define better classical algorithms, but intrinsic quantum properties are not used. Quantum inspired algorithms are out of the scope of this work.

1.4 Research road map and main contributions

This work research road map is displayed in Figure 1. As a result of this research work the following contributions can be enumerated.

1. An open-ended research question about learning in quantum neural networks.
2. We show that a previous quantum neural network model can be efficiently simulated by a classical single layer neural network and its learning algorithm is not a quantum learning algorithm.

Figure 1 – Research road map showing the goal and contributions of the thesis



Source: author

3. One formalization and definition of the quantum neural networks general concept. This concept can guide the development of a quantum neural network model.

4. Definition of a quantum weightless neural network model and its learning algorithm.

5. Definition of a quantum perceptron named quantum perceptron over a field. This neuron can be considered as a direct generalization of a classical perceptron to the quantum field.

6. Definition of a nonlinear quantum learning algorithm for neural networks that can select neural networks free parameters and architecture in linear time. The proposed learning algorithm is the first quantum algorithm that can determine the best neural network architecture (with best accuracy in the training set) in linear time.

   During the development of this thesis some unrelated contributions where obtained.

7. Inspired by quantum computing we show how a weightless neural network can be represented by a matrix of sufficient generality as to have classical weighted, classical weightless (RAM-based, PLN, etc), quantum weighted and quantum weightless neural models as particular cases.

8. Inspired by quantum computing we show that is possible to simulate a Probabilistic Automata (PA) with a single layer Weightless Neural Network (WNN) with no auxiliary data structures.

1.5 Bibliographical production

In this Section the list of research papers submitted/published by Adenilton J. da Silva during his doctoral work at the Centro de Informática, UFPE is presented.

1.5.1 Articles in scientific journals

1. **da Silva, Adenilton J.**; de Oliveira, Wilson R.; Ludermir, Teresa B. (2015) *Comments on "quantum M-P neural networks"*, International Journal of Theoretical Physics, v. 54(6), pp. 1878-1881

2. **da Silva, Adenilton J.**; de Oliveira, Wilson R.; Ludermir, Teresa B. (2012). *Classical and superposed learning in quantum weightless neural networks*. Neurocomputing. v. 75(1), pp. 52-60

3. **da Silva, Adenilton J.**; de Oliveira, Wilson R.; Ludermir, Teresa B. (2015). *Weightless neural network parameters and architecture selection in a quantum computer*. Neurocomputing (article accepted for publication)

4. **da Silva, Adenilton J.**; Ludermir, Teresa B.; de Oliveira, Wilson R. *Quantum perceptron over a field* (submitted to International Journal of Neural Systems).

5. de Paula Neto, F. M.; de Oliveira, Wilson R.; **da Silva, Adenilton J.**; Ludermir, Teresa B. . *Chaos in Quantum Weightless Neuron Node Dynamics* (article accepted to publication in Neurocomputing)

   Presents the analysis of a quantum weightless neuron dynamics. Results of this work are not displayed in this Thesis.

6. de Lima, Tiago P., **da Silva, Adenilton J.**, Ludermir, Teresa B., de Oliveira, Wilson R. (2014). An automatic methodology for construction of multi-classifier systems based on the combination of selection and fusion. Progress in Artificial Intelligence, 2(4), 205-215.

Presents a methodology for the automatic construction of multi-classifiers systems. Results of this work are not displayed in this Thesis.

1.5.2 Complete works published in proceedings of conferences

1. de Paula Neto, F. M. ; Ludermir, Teresa B. ; de Oliveira, Wilson R. ; **da Silva, Adenilton J.** . Resolvendo 3-SAT com Neurônios Quânticos Sem Peso. In: V Workshop-Escola de Computação e Informação Quântica. V Workshop-Escola de Computação e Informação Quântica, 2015.

2. **da Silva, Adenilton J.** ; de Oliveira, Wilson R. ; Ludermir, Teresa B. . Training a classical weightless neural network in a quantum computer. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges 2014. p. 523-528.

3. **da Silva, Adenilton J.** ; de Oliveira, Wilson R. ; Ludermir, Teresa B. . Probabilistic automata simulation with single layer weightless neural networks. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, 2014. p. 547-552.

4. de Oliveira, Wilson R. ; **da Silva, Adenilton J.** ; Ludermir, Teresa B. . Vector space weightless neural networks. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, 2014. p. 535-540.

5. de Paula, F. M. ; **da Silva, Adenilton J.** ; Ludermir, Teresa B.; de Oliveira, Wilson R. . Analysis of Quantum Neural Models. In: Congresso Brasileiro de Inteligência computacional, 2013, Recife.

6. **da Silva, Adenilton J.** ; Ludermir, Teresa B. ; de Oliveira, Wilson R. Single-shot learning algorithm for quantum weightless neural networks. In: Congresso Brasileiro de Inteligência computacional, 2013, Recife. Congresso brasileiro de inteligência computacional, 2013.

7. **da Silva, Adenilton J.** ; Ludermir, Teresa B. ; de Oliveira, Wilson R. On the Universality of Quantum Logical Neural Networks. In: 2012 Brazilian Symposium on Neural Networks (SBRN), 2012, Curitiba.p. 102-106.

8. Mineu, Nicole L. ; **da Silva, Adenilton J.** ; Ludermir, Teresa B. . Evolving Neural Networks Using Differential Evolution with Neighborhood-Based Mutation and Simple Subpopulation Scheme. In: 2012 Brazilian Symposium on Neural Networks (SBRN), 2012, Curitiba. p. 190-195.

9. Lima, Tiago P. F. ; **da Silva, Adenilton J.** ; Ludermir, Teresa B. .Clustering and Selection of Neural Networks Using Adaptive Differential Evolution. In: International Joint Conference on Neural Network, 2012, Brisbane. Proceedings of IJCNN 2012. Los Alamitos: IEEE Computer Society, 2012

10. Lima, Tiago P. F. ; **da Silva, Adenilton J.** ; Ludermir, Teresa B. . Selection and Fusion of Neural Networks via Differential Evolution. Advances in Artificial Intelligence – IBERAMIA 2012. Springer Berlin Heidelberg, 2012, v. 7637, p. 149-158.

1.6 Outline

This Chapter is a Sumary article linking the other Parts of this Thesis and the conclusion is displayed in Section 1.7. The remainder of this Thesis is divided in 4 parts. Each part consists of a published, accepted or submitted article about quantum neural networks.

Part I analyses a previously proposed quantum neural network model named quantum M-P neural network (QMPN). It is shown an equivalence between the quantum M-P neural network (ZHOU; DING, 2007) and a classical perceptron and it is shown that the QMPN learning algorithm is not a quantum algorithm.

Part II proposes a quantum neural network named qRAM and its learning algorithm. Weightless neural networks were chosen for its simplicity which allows direct quantization. The qRAM neural networks can simulate the classical weightless neural networks models and can be trained with a classical or quantum algorithm.

Part III extends results presented in Part I and deals with neural network architecture selection in the context of weightless neural networks. It is shown that neural network architecture selection can be performed through the training of a quantum weightless neural network.

Part IV deals with architecture selection in the context of weighted neural networks. A model of quantum perceptron named quantum perceptron over a field is defined, and a linear time learning algorithm for architeture selection of perceptrons is presented. The proposed learning algorithm optimizes weights and architectures simultaneously.

Section 1.7 is the Conclusion. Subsection 1.7.1 presents the main results of this Thesis. Subsection 1.7.2 displays some directions in the research on quantum neural networks. Finally Subection 1.7.3 shows some concluding remarks.

1.7 Conclusion

This Thesis dealt with the question of design and training of a neural network using a quantum computer. Quantum neural computing is a new field with many challenges. The main question of this work is about neural network architecture selection using quantum techniques. This question naturally arose from previous works on neural network architecture selection (DA SILVA; MINEU; LUDERMIR, 2010) and quantum neural networks (DE OLIVEIRA et al., 2008). This Section presents the main results of this thesis, analyses the limitations of the proposed model and algorithm and presents some possible future works.

1.7.1 Main results

In (ZHOU; DING, 2007) a quantum neural network named Quantum M-P Neural Network (QMP) is proposed. Part I showed that the QMPN algorithm is not a quantum algorithm and that QMPN neurons can be efficiently simulated by a single layer classical neural network. Part I is the article (DA SILVA; DE OLIVEIRA; LUDERMIR, 2014a) published in the International Journal of Theoretical Physics.

The first result of this thesis was the definition of a quantum weightless neural network. This step was important because previously proposed quantum neural networks present some problems, for instance in (ZHOU; DING, 2007) where the quantum neuron can enter in a non unitary configuration. The description of the quantum weightless neural network and its learning algorithm is described in Part II that presented the article (DA SILVA; DE OLIVEIRA; LUDERMIR, 2012) published in the Neurocomputing Journal.

Part III showed how to perform artificial neural network architecture selection through the learning procedure of a qRAM weightless neural network. This strategy is the first algorithm proposed to perform neural network architecture selection in a quantum computer. Part III is an accepted article in Neurocomputing Journal.

Part IV extends the results of Parts II and III to a neural network with weights. A quantum perceptron named Quantum Perceptron over a Field is proposed. The QPF differs from previous quantum neural networks since it can be viewed as a direct generalisation of the classical perceptron, can be trained by a quantum learning algorithm, cannot enter in a non unitary configuration and have well defined inner operations. QPF is the first quantum perceptron that have all these properties together.

The architecture of a multilayer quantum perceptron over a field can be represented in a quantum computer as a matrix over field F. This representation allows any number of layers and neurons by layers and do not break any quantum computing principle.

A multilayer QPF corresponds to a fixed quantum operator and its configuration is represented by a string of qubits.

In (PANELLA; MARTINELLI, 2011; PANELLA; MARTINELLI, 2009; DA SILVA; DE OLIVEIRA; LUDERMIR, 2012) the training procedure of a neural network uses a nonlinear quantum operator and the superposition of all neural networks configurations for a given architecture. In this Thesis we extend the idea in (PANELLA; MARTINELLI, 2011) and we defined a learning algorithm named Superposition based Architecture Learning algorithm (SAL) that performs a non-linear search in the neural network parameters and the architecture space simultaneously. The main advantage of our learning algorithm and previously proposed learning algorithm is its ability to perform a global search in the space of weights and architecture with linear cost in the number of examples in the training set and in the number of bits used to represent the neural network. This result depends of two suppositions: i) the existence of a quantum computer and ii) a nonlinear quantum operator proposed in (ABRAMS; LLOYD, 1998). The main conclusion about architecture selection is that with the supposition of nonlinear quantum computing we can select an optimal neural network architecture in linear time. All results about the QPF and its learning algorithm were detailed in Part IV that is an article submitted to the International Journal of Neural Systems.

There is no guarantee in the viability of nonlinear quantum computing and quantum operators. The existence of a unitary quantum algorithm that performs neural network architecture selection exploiting superposition as suggested in the framework presented in Fig. 2 of Part IV with better computational time than the classical algorithms is an open problem. Superposition of quantum neural networks can allow architecture evaluation in a way that is not possible in classical neural computing.

1.7.2 Future works

Several future directions can be derived from this work. Some of the possibilities are described below. Next subsections points directions to perform some of these future works.

- To develop a linear version of the superposition architecture based learning algorithm.
- To analyse the properties of quantum neural networks based on quantum associative memories using benchmark datasets.
- To study the computational power of the quantum neural networks.
- To analyse the possibility of use classical methods to train a quantum neural model.
- Analyse the possibility of training an ensemble of neural networks in superposition.

1.7.2.1 Quantum linear architecture learning algorithm

The final step of the Quantum architecture selection learning algorithm in Part IV is a non-linear search in the architecture and weights space. In this step, free parameters will collapse to a basis state not in superposition. One possible future work is to analyse how one can use the neural network with weights in superposition. In this way one could take advantage of superposition in a trained neural network.

In its first steps, SAL algorithm initialises the neural network weights with a superposition of all possible weights for some different architectures. The result of this initialisation is a superposition of a finite number of neural networks. This superposition can be interpreted as a superposition of untrained neural networks. One possible future work is to perform a quantum procedure to train these neural networks in superposition and to perform quantum search only in the trained neural networks.

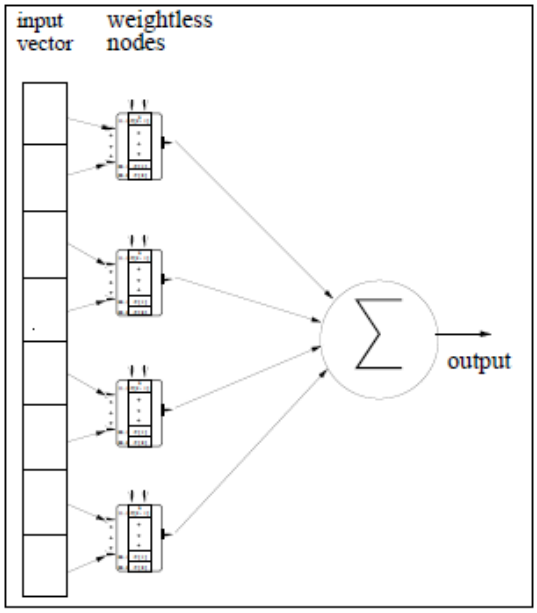
Any classical computation can be simulated by a quantum computer. In (DE OLIVEIRA et al., 2008; DA SILVA; DE OLIVEIRA; LUDERMIR, 2012) some classical learning algorithms are adapted to work in a quantum computer. However algorithms as backpropagations do not have a known quantisation.

1.7.2.2 Weightless neural networks in quantum memories

A RAM node is a small piece of RAM memory and is used in several applications. There are several proposals for quantum memories (VENTURA; MARTINEZ, 2000; TRUGENBERGER, 2001; TRUGENBERGER, 2002) and neural networks models based on quantum memories (DA SILVA; DE OLIVEIRA; LUDERMIR, 2010; ZHOU, 2010; ANDRECUT; ALI, 2002). However, there is not an analysis of the development of these quantum neural networks using the architecture of a weightless neural network. For instance, there is no analysis if a pyramidal or WISARd architecture can be efficiently used. A WISARd (LUDERMIR et al., 1999) is composed of components named discriminators. A discriminator is displayed in Figure 2, each weightless node receives few inputs and their outputs are summed to determine the pertinence of the input example to the class associated with the discriminator. Weightless nodes in the discriminator have a small number of inputs.

The study of small pieces of quantum memories as neurons of a neural network may possibly allow the application of quantum computing using only small quantum processors that are actually created in physical experiments. Using neurons with low dimensionality, simulation of this kind of quantum neural network will be possible in classical computers.

Figure 2 – Discriminator

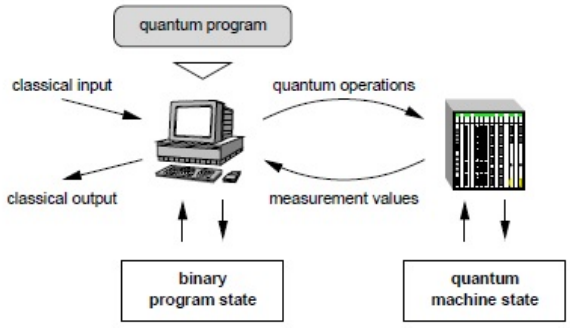


Source: Ludermir et al. (1999)

1.7.2.3 Classical learning in quantum models

Heterogeneous computer architectures with different processors specialised for different tasks are increasingly common. One possible future architecture is a computer where the quantum processor is a host device controlled by a classical main device. This scenario is displayed in Figure 3.

Figure 3 – Quantum computer with hybrid architecture



Source: Ömer (2005)

Hybrid quantum algorithms may allow the development of new heuristic algorithms effectively using quantum computation. A programming language for hybrid quantum computers is proposed in (ÖMER, 2005). With this kind of architecture, classical methods

maybe can be used to control the quantum system using a measure and feedback strategy. For instance, the quantum system can implement a neural network that is used as the evaluation function of a classical evolutionary algorithm.

1.7.2.4 Computability of quantum neural network models

Computability of artificial neural networks has been studied in several works (SIEGELMANN; SONTAG, 1991; CABESSA; SIEGELMANN, 2014; LUDERMIR, 1992; SOUTO; LUDERMIR; CAMPOS, 2000; DE OLIVEIRA; DE SOUTO; LUDERMIR, 2003; DA SILVA; DE OLIVEIRA; LUDERMIR, 2014b). In (LUDERMIR, 1992) and (SOUTO; LUDERMIR; CAMPOS, 2000) authors showed an equivalence between weightless neural networks and probabilistic automata. Probabilistic automata (RABIN, 1963) are more powerful than finite state automata (HOPCROFT; MOTWANI; ULLMAN, 2007) and can recognize all regular languages, some context-free, some context-sensitive and some recursive enumerable languages, cutting the Chomsky Hierarchy (HOPCROFT; MOTWANI; ULLMAN, 2007) elliptically. In (DE OLIVEIRA; DE SOUTO; LUDERMIR, 2003) is presented how to simulate a Turing Machine with a RAM neural network and an additional data structure.

In (SIEGELMANN; SONTAG, 1991) the complexity of recurrent neural networks is analysed. Recurrent neural networks with rational weights can recognise any recursively enumerable language and weighted neural networks with real weights has super Turing capabilities. Auto adaptive neural networks modes can also recognise more than recursively enumerable languages as show in (CABESSA; SIEGELMANN, 2014).

One possible future work is to verify the computational power of the quantum neural networks previously proposed in literature. For instance, matrix defined quantum perceptron as the ZQP can approximate any quantum operator and can be considered as a universal model of quantum computing. With fixed operator quantum neural networks, we cannot guarantee this universality and further studies are necessary to determine the computational power of the quantum perceptron over a field and of previously proposed fixed operator quantum neural networks.

1.7.3 Concluding remarks

The advent of quantum computation dramatically change the way that computers are programmed and allow the solution of some problems more efficiently than in a classical computer. Quantum algorithms that overcome classical algorithms use the concept of quantum parallelism. The path that computer scientist need to follow to develop quantum algorithms is hard because computer scientists do not known quantum principles.

This thesis proposes a framework to select a neural network architecture in a

quantum computer. The main result is that the selection of a neural network architecture can be performed in a nonlinear quantum computer with linear time in relation to the number of example in the training set and the number of bits used in the neural network representation. The proposed framework is the first strategy proposed to perform neural network architecture selection in a quantum computer. This idea originated from our previous works on architecture selection and quantum neural networks.

This thesis presents the results of a theoretical work that seek to unify two different concepts related to the learning of neural networks: automatic architecture selection and quantum learning algorithms. The combination of these concepts brings advantages in the selection of neural networks architectures because the concept of quantum parallelism allow a better evaluation of neural network architectures. The long term plan is to develop a quantum learning algorithm for architecture selection without nonlinear quantum operators. Such linear algorithm can allow architecture selection in quantum computers as soon as a quantum computer is available.

References

ABRAHAM, A. Meta learning evolutionary artificial neural networks. *Neurocomputing*, v. 56, p. 1 – 38, 2004.

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.

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. 7–9, p. 1438 – 1450, 2010.

ALTAISKY, M.; KAPUTKINA, N.; KRYLOV, V. Quantum neural networks: Current status and prospects for development. *Physics of Particles and Nuclei*, v. 45, n. 6, p. 1013–1032, 2014.

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

ANDRECUT, M.; ALI, M. K. a Quantum Neural Network Model. *International Journal of Modern Physics C*, v. 13, n. 01, p. 75–88, 2002.

BEHRMAN, E. C. et al. Simulations of quantum neural networks. *Information Sciences*, v. 128, n. 3-4, p. 257–269, 2000.

BEIGY, H.; MEYBODI, M. R. Backpropagation algorithm adaptation parameters using learning automata. *International Journal of Neural Systems*, v. 11, n. 03, p. 219–228, 2001. PMID: 11574959. Disponível em: <http://www.worldscientific.com/doi/abs/10.1142/S0129065701000655>.

BENIOFF, P. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, Kluwer Academic Publishers-Plenum Publishers, v. 22, n. 5, p. 563–591, 1980. ISSN 0022-4715. Disponível em: <http://dx.doi.org/10.1007/BF01011339>.

BIAN, Z. et al. Experimental determination of ramsey numbers. *Physical Review Letters*, v. 111, 2013.

CABESSA, J.; SIEGELMANN, H. T. The super-turing computational power of plastic recurrent neural networks. *International Journal of Neural Systems*, v. 24, n. 08, p. 1450029, 2014.

DA CRUZ, A. V.; VELLASCO, M. M.; PACHECO, M. A. Quantum-inspired evolutionary algorithm for numerical optimization. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. [S.l.: s.n.], 2006. p. 2630–2637.

DA SILVA, A. J.; DE OLIVEIRA, W. R.; LUDERMIR, T. B. A weightless neural node based on a probabilistic quantum memory. In: IEEE. *Neural Networks (SBRN), 2010 Eleventh Brazilian Symposium on*. [S.l.], 2010. p. 259–264.

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

DA SILVA, A. J.; DE OLIVEIRA, W. R.; LUDERMIR, T. B. Comments on "quantum m-p neural network". *International Journal of Theoretical Physics*, Springer US, p. 1–4, 2014. ISSN 0020-7748. Disponível em: <http://dx.doi.org/10.1007/s10773-014-2393-1>.

DA SILVA, A. J.; DE OLIVEIRA, W. R.; LUDERMIR, T. B. Probabilistic automata simulation with single layer weightless neural networks. In: *European Symposium on Artificial Neural Network*. [S.l.: s.n.], 2014. p. 547 –552.

DA SILVA, A. J.; MINEU, N. L.; LUDERMIR, T. B. Evolving artificial neural networks using adaptive differential evolution. In: *Advances in Artificial Intelligence - IBERAMIA 2010*. [S.l.]: Springer Berlin Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6433). p. 396–405.

DE OLIVEIRA, W.; DE SOUTO, M. C. P.; LUDERMIR, T. Turing's analysis of computation and artificial neural networks. *Journal of Intelligent & Fuzzy Systems*, v. 13, n. 2-4, p. 85–98, 2003.

DE OLIVEIRA, W. R. et al. Quantum logical neural networks. In: *Brazilian Symposium on Neural Networks*. [S.l.: s.n.], 2008. p. 147 –152.

DEUTSCH, D. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, The Royal Society, v. 400, n. 1818, p. 97–117, 1985.

DUAN, H.; XU, C.; XING, Z.-H. A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. *International Journal of Neural Systems*, v. 20, n. 1, p. 39–50, 2010.

DUAN, L. et al. Long-distance quantum communication with atomic ensembles and linear optics. *Nature*, v. 414, n. 6862, p. 413–418, 2001.

FARHI, E.; GUTMANN, S. Quantum computation and decision trees. *Phys. Rev. A*, v. 58, p. 915–928, Aug 1998.

FEYNMAN, R. Simulating physics with computers. *International Journal of Theoretical Physics*, v. 21, p. 467, 1982.

GARRIDO, C.; DE OÑA, R.; DE OÑA, J. Neural networks for analyzing service quality in public transportation. *Expert Systems with Applications*, v. 41, p. 6830–6838, 2014.

GEORGESCU, I. Foundations of quantum mechanics. *Nature Physics*, v. 10, n. 4, p. 253–253, 2014.

GONZÁLEZ, M. et al. Retrieval of noisy fingerprint patterns using metric attractor networks. *International Journal of Neural Systems*, v. 24, n. 07, p. 1450025, 2014.

GROVER, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, v. 79, p. 325–328, 1997.

HAN, K.-H.; KIM, J.-H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *Evolutionary Computation, IEEE Transactions on*, IEEE, v. 6, n. 6, p. 580–593, 2002.

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

HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. D. *Introduction to Automata Theory, Languages and Computation*. 3. ed. [S.l.]: Pearson Addison-Wesley, 2007. ISBN 978-0-321-51448-6.

HSU, W.-Y. Application of quantum-behaved particle swarm optimization to motor imagery eeg classification. *International journal of neural systems*, v. 23, n. 6, p. 1350026, 2013.

JENNEWEIN, T. et al. A fast and compact quantum random number generator. *Review of Scientific Instruments*, v. 71, n. 4, p. 1675–1680, 2000.

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

KOCH, C.; HEPP, K. Quantum mechanics in the brain. *Nature*, v. 440, n. 7084, p. 611–611, 2006.

KORDÍK, P. et al. Meta-learning approach to neural network optimization. *Neural Networks*, v. 23, n. 4, p. 568 – 582, 2010.

KOUDA, N. et al. Qubit neural network and its learning efficiency. *Neural Comput. Appl.*, v. 14, n. 2, p. 114–121, 2005.

KOURENTZES, N.; BARROW, D. K.; CRONE, S. F. Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, v. 41, p. 4235–4244, 2014. ISSN 09574174.

KOZA, J. R.; RICE, J. P. Genetic generation of both the weights and architecture for a neural network. In: *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*. [S.l.: s.n.], 1991. ii, p. 397–404.

LI, P. et al. A hybrid quantum-inspired neural networks with sequence inputs. *Neurocomputing*, v. 117, p. 81 – 90, 2013.

LIMA, T. P.; LUDERMIR, T. B. An automatic method for construction of ensembles to time series prediction. *International Journal of Hybrid Intelligent Systems*, v. 10, n. 4, p. 191–203, 2013.

LUDERMIR, T. Computability of logical neural networks. *Journal of Intelligent Systems*, v. 2, n. 1, p. 261–290, 1992.

LUDERMIR, T. B. et al. Weightless neural models: a review of current and past works. *Neural Computing Surveys*, v. 2, p. 41–61, 1999.

LUDERMIR, T. B.; DE OLIVEIRA, W. R. Particle Swarm Optimization of MLP for the identification of factors related to Common Mental Disorders. *Expert Systems with Applications*, v. 40, p. 4648–4652, 2013.

MACARIO, V. et al. Metaclasses and zoning for handwritten document recognition. In: *Neural Networks (IJCNN), The 2013 International Joint Conference on*. [S.l.: s.n.], 2013. p. 1–5.

MALOSSINI, A.; CALARCO, T. Quantum genetic optimization. *IEEE Transactions on Evolutionary Computation*, v. 12, p. 231–241, 2008.

MATTOS NETO, P. S. et al. Hybrid intelligent system for air quality forecasting using phase adjustment. *Engineering Applications of Artificial Intelligence*, v. 32, p. 185 – 191, 2014.

MICROSOFT. *Quantum Computing - Creating a new generation of computing devices*. 2014. <http://research.microsoft.com/en-us/about/our-research/quantum-computing.aspx>. Accessed: 4-January-2015.

MONZ, T. et al. 14-qubit entanglement: Creation and coherence. *Physical Review Letters*, v. 106, 2011.

MORET-BONILLO, V. Can artificial intelligence benefit from quantum computing? *Progress in Artificial Intelligence*, p. 1–17, 2014.

NARAYANAN, A.; MENNEER, T. Quantum artificial neural networks architectures and components. *Information Sciences*, v. 128, n. 3-4, p. 231–255, 2000.

NASA. *Quantum Artificial Intelligence Laboratory*. 2013. <http://www.nas.nasa.gov/quantum/index.html>. Accessed: 4-January-2015.

NEVEN, H. *Launching the Quantum Artificial Intelligence Lab*. 2013. <http://googleresearch.blogspot.pt/2013/05/launching-quantum-artificial.html>. Accessed: 4-January-2015.

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

ÖMER, B. Classical concepts in quantum programming. *International Journal of Theoretical Physics*, v. 44, n. 7, p. 943–955, 2005.

PACIFICO, L.; LUDERMIR, T. Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies. In: *Neural Networks (IJCNN), The 2013 International Joint Conference on*. [S.l.: s.n.], 2013. p. 1–6.

PANELLA, M.; MARTINELLI, G. Neurofuzzy networks with nonlinear quantum learning. *Fuzzy Systems, IEEE Transactions on*, IEEE, v. 17, n. 3, p. 698–710, 2009.

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.

PENROSE, R. *The emperor's new mind: concerning computers, minds, and the laws of physics*. [S.l.]: Oxford University Press, 1999.

RABIN, M. O. Probabilistic automata. *Information and control*, v. 6, n. 3, p. 230–245, 1963.

REBENTROST, P.; MOHSENI, M.; LLOYD, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, v. 113, p. 130503, 2014.

RICKS, B.; VENTURA, D. Training a quantum neural network. In: *Advances in Neural Information Processing Systems*. Cambridge, MA: [s.n.], 2004.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. Quantum walks on graphs representing the firing patterns of a quantum neural network. *Physical Review A - Atomic, Molecular, and Optical Physics*, v. 89, 2014.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. The quest for a Quantum Neural Network. *Quantum Information Processing*, 2014.

SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, v. 26, n. 5, p. 1484–1509, 1997.

SHOR, P. W. Why haven't more quantum algorithms been found? *J. ACM*, v. 50, n. 1, p. 87–90, 2003.

SIEGELMANN, H. T.; SONTAG, E. D. Turing computability with neural nets. *Applied Mathematics Letters*, Elsevier, v. 4, n. 6, p. 77–80, 1991.

SOUTO, M. C. P. de; LUDERMIR, T.; CAMPOS, M. Encoding of probabilistic automata into ram-based neural networks. In: *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*. [S.l.: s.n.], 2000. v. 3, p. 439–444 vol.3.

STAFFA, M. et al. Can you follow that guy? In: *European Symposium on Artificial Neural Networks*. [S.l.: s.n.], 2014. p. 511–516.

STRACHAN, G. et al. Quantum-inspired multi-gene linear genetic programming model for regression problems. In: *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*. [S.l.: s.n.], 2014. p. 152–157.

TEGMARK, M. Why the brain is probably not a quantum computer. *Information sciences*, v. 128, p. 155–179, 2000.

TICKNOR, J. L. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, v. 40, p. 5501–5506, 2013.

TIRUMALA, S. S.; CHEN, G.; PANG, S. Quantum inspired evolutionary algorithm by representing candidate solution as normal distribution. In: *Neural Information Processing*. [S.l.: s.n.], 2014. p. 308–316.

TRUGENBERGER, C. A. Probabilistic quantum memories. *Phys. Rev. Lett.*, v. 87, n. 6, p. 067901, 2001.

TRUGENBERGER, C. A. Quantum pattern recognition. *Quantum Information Processing*, v. 1, p. 471–493, 2002.

VENTURA, D.; MARTINEZ, T. Initializing the Amplitude Distribution of a Quantum State. *Journal of Genetic Counseling*, v. 12, n. 6, p. 547–559, 1999.

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

WIEBE, N.; KAPOOR, A.; SVORE, K. M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum information & computation*, v. 15, n. 3&4, p. 316–356, 2015.

YAMAZAKI, A.; LUDERMIR, T. B. Neural network training with global optimization techniques. *International Journal of Neural Systems*, v. 13, n. 02, p. 77–86, 2003. PMID: 12923920. Disponível em: <http://www.worldscientific.com/doi/abs/10.1142/S0129065703001467>.

ZHANG, G. et al. An optimization spiking neural p system for approximately solving combinatorial optimization problems. *International Journal of Neural Systems*, v. 24, n. 05, p. 1440006, 2014.

ZHOU, J. et al. Recognition of handwritten numerals by quantum neural network with fuzzy features. *International Journal on Document Analysis and Recognition*, v. 2, n. 1, p. 30–36, 1999.

ZHOU, R. Quantum competitive neural network. *International Journal of Theoretical Physics*, Springer, v. 49, n. 1, p. 110–119, 2010.

ZHOU, R.; DING, Q. Quantum m-p neural network. *International Journal of Theoretical Physics*, v. 46, p. 3209–3215, 2007.

ZHOU, R.; QIN, L.; JIANG, N. Quantum perceptron network. In: *Artificial Neural Networks–ICANN 2006*. [S.l.: s.n.], 2006. v. 4131, p. 651–657.

Part I

Comments on "Quantum M-P Neural Network"

Int J Theor Phys (2015) 54:1878–1881
DOI 10.1007/s10773-014-2393-1

Comments on "Quantum M-P Neural Network"

Adenilton J. da Silva · Wilson R. de Oliveira · Teresa B. Ludermir

Received: 12 June 2014 / Accepted: 23 October 2014 / Published online: 19 November 2014

Abstract In a paper on quantum neural networks, Zhou and Ding (Int. J. Theor. Phys. 46(12):3209-3215 (2007)) proposed a new model of quantum perceptron denoted quantum M-P neural network and showed its functionality by an example. In this letter, we show that the proposed learning algorithm does not follow an unitary evolution and the proposed neuron can be efficiently simulated by a classical single layer neural network.

Keywords Quantum computation · Neural networks · Quantum learning

1 Introduction

In [2] Kak proposed an idea of quantum neural computation. Since then, several researchers have proposed quantum neural networks [6–8] and quantum inspired neural networks [3, 4]. In [8] Zhou and Ding proposed a quantum perceptron; they called the new neuron model as quantum M-P neural network (qMPN). The weights of qMPN are stored in a squared matrix W. Let $|x\rangle$ be an input vector, the output $|y\rangle$ can be calculated as in (1).

$$|y\rangle = W|x\rangle \tag{1}$$

Representing a neural network as a quantum operator can bring new possibilities to neural computation. A quantum neural network can receive any superposition of quantum states

A. J. da Silva (✉) · T. B. Ludermir
Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil
e-mail: ajs@deinfo.ufrpe.br

T. B. Ludermir
e-mail: tbl@cin.ufpe.br

A. J. da Silva · W. R. de Oliveira
Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco, Recife, Brazil

W. R. de Oliveira
e-mail: wilson.rosa@gmail.com

at its inputs which can be seen as untrained data. The network operator acts linearly. For instance, if a neuron with weight matrix W receives the input $|\alpha\rangle = \sum_{k=1}^{n} \alpha_k |k\rangle$ it will act linearly in each value in the superposition and its output will be $|y\rangle = \sum_{k=1}^{n} \alpha_k W|k\rangle$.

In Section 4.2 of [8] is presented a prepocessing step to work with non-orthogonal states. This prepossessing step changes the input representation from qubits (or vectors) to a matrix of inner products. Any quantum bit can be represented as a superposition of orthogonal vectors and this step is not used in any other section of [8]. This prepocessing is a nonquantum operation for it accesses the amplitudes since inner product is employed freely. Calculating the inner product with a basis element results the amplitude corresponding to that basis element in a superposition.

In this letter we first show that the proposed learning algorithm for qMPN does not preserve unitary operators and it can produce non unitary weight matrices. After these steps we show that any qMPN can be efficiently simulated by a single layer neural network composed of classical perceptrons and that the learning algorithm of the qMPN is exactly the classical perceptron learning rule.

2 Learning Algorithm

The learning algorithm proposed in [8] for qMPN is described in Algorithm 1. The weights update rule of Algorithm 1 is described in (2), where w_{ij} are the entries of the $n \times n$ matrix W, $1 \leq i, j, \leq n$, η is a learning rate, $|d\rangle_i$ and $|y\rangle_i$ corresponds to the ith probability amplitude of n-dimensional qubits $|d\rangle$ and $|y\rangle$, and $|x\rangle_j$ is the jth probability amplitude of the n-dimensional qubit $|x\rangle$.

Algorithm 1 Learning algorithm qMPN

1 Let $W(0)$ be a weight matrix
2 Given a set of quantum examples in the form $(|x\rangle, |d\rangle)$, where $|x\rangle$ is an input and $|d\rangle$ is the desired output
3 Calculate $|y\rangle = W(t)|x\rangle$, where t is the iteration number
4 Update the weights following the learning rule described in (2).
5 Repeat steps 3 and 4 until a stop criterion is met.

$$w_{ij}(t+1) = w_{ij}(t) + \eta \left(|d\rangle_i - |y\rangle_i\right) |x\rangle_j \tag{2}$$

If $|d\rangle = \alpha|0\rangle + \beta|1\rangle$, then $|d\rangle_1 = \alpha$ and $|d\rangle_2 = \beta$ are the probability amplitudes of the state $|d\rangle$. In quantum computation, one cannot direct access a probability amplitude [5]. Measuring a qubit returns only $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$. Therefore, the learning rule described in (2) is not a quantum learning rule and one cannot use a quantum computer to train the qMPN with Algorithm 1.

One alternative is to use a classical computer to train the qMPN with Algorithm 1. However this strategy can lead to non-unitary neurons configurations, as we show in (3), where $W(0) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|d\rangle = \frac{1}{\sqrt{2}}(|0\rangle + 1)$, and $\eta = 1$. After the first learning algorithm iteration the qMPN will be represented by the matrix $W(1)$. Clearly $W(1)$ is a non unitary operator. Quantum gates must be unitary operators [5], so the qMPN

trained with this algorithm is not necessarily a quantum neuron. We conclude this paragraph with Theorem 1.

$$
\begin{aligned}
|y\rangle = W(0)|x\rangle &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\
w_{00}(t+1) &= w_{00}(t) + \eta \left(|d\rangle_0 - |y\rangle_0 \right) |x\rangle_0 = \\
&= 0 + \left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} \right) \cdot \frac{1}{\sqrt{2}} = 0 \\
w_{01}(t+1) &= w_{01}(t) + \eta \left(|d\rangle_0 - |y\rangle_0 \right) |x\rangle_1 = \\
&= 1 + \left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} \right) \cdot \left(\frac{1}{\sqrt{2}} \right) = 1 \\
w_{10}(t+1) &= w_{10}(t) + \eta \left(|d\rangle_1 - |y\rangle_1 \right) |x\rangle_0 = \\
&= -1 + \left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \right) \cdot \left(\frac{1}{\sqrt{2}} \right) = 0 \\
w_{11}(t+1) &= w_{11}(t) + \eta \left(|d\rangle_1 - |y\rangle_1 \right) |x\rangle_1 = \\
&= 0 + \left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \right) \cdot \left(\frac{1}{\sqrt{2}} \right) = 1 \\
W(1) &= \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3}
$$

Theorem 1 *qMPN learning rule does not preserve unitary operators.*

We verified that Algorithm 1 is not a quantum algorithm. The main question in this letter is about the differences between the perceptrons and the qMPN and if the Algorithm 1 presents some advantage when compared with classical neural networks learning algorithm. Before start this analysis we define a classical neuron.

A neuron with inputs $x_1, x_2, \cdots, x_n$ and weights $w_1, w_2, \cdots, w_n$ and linear activation function f has its output y described in (4). One possible weight update rule for an artificial neuron is the Least Mean Square (LMS) rule [1] described in (5).

$$
y = f\left(\sum_{i=1}^{n} x_i \cdot w_i \right) \tag{4}
$$

$$
w_i(t+1) = w_i(t) + \eta \cdot (d - y) \cdot x_i \tag{5}
$$

Now we present an example with two artificial neurons, where the weights of the first neuron are w_{11} and w_{12} and the weights of the second neuron are w_{21} and w_{22}. If the neurons receive an input $\begin{bmatrix} x_1 & x_2 \end{bmatrix}$, then the output of the first neuron will be $y_1 = x_1 w_{11} + x_2 w_{12}$ and the output of the second neuron will be $y_2 = x_1 w_{21} + x_2 w_{22}$.

One can organize the weights in a matrix $W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$, the inputs in a vector $|x\rangle = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, and the neurons outputs can be calculated exactly as in (1), where $|y\rangle = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$. The desired outputs for $|x\rangle$ can be represented with a vector $|d\rangle = \begin{bmatrix} d_1 & d_2 \end{bmatrix}$, where d_i is the desired output of ith neuron, when $|x\rangle$ is presented. We rewrite (5) using a matrix notation and we obtain exactly the learning rule in (2) that was proposed in [8].

The authors also shown the capacity of the qMPN to solve non linearly separable patterns. But to perform this task the values 00, 01, 10, 11 were associated with qubits in the computational basis that are linearly independents.

3 Conclusion

We conclude claiming that for any qMPN one can create an equivalent classical single layer neural network. We also verified that the learning algorithm proposed by Zhou and Ding does not present any advantage over the classical ones and works exactly as the LMS rule.

Acknowledgments This work is supported by research grants from CNPq, CAPES and FACEPE (Brazilian research agencies).

References

1. Haykin, S.: Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, Upper Saddle River, NJ (1999)
2. Kak, S.C.: On quantum neural computing. Inf. Sci. **83**(3), 143–160 (1995)
3. Kouda, N., Matsui, N., Nishimura, H., Peper, F.: Qubit neural network and its learning efficiency. Neural Comput. Appl. **14**(2), 114–121 (2005)
4. Li, P., Xiao, H., Shang, F., Tong, X., Li, X., Cao, M.: A hybrid quantum-inspired neural networks with sequence inputs. Neurocomputing **117**, 81–90 (2013)
5. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
6. Panella, M., Martinelli, G.: Neural networks with quantum architecture and quantum learning. Int. J. Circ. Theor. Appl. **39**, 61–77 (2011). doi:10.1002/cta.619
7. da Silva, A.J., de Oliveira, W.R., Ludermir, T.B.: Classical and superposed learning for quantum weightless neural networks. Neurocomputing **75**(1), 52–60 (2012)
8. Zhou, R., Ding, Q.: Quantum m-p neural network. Int. J. Theor. Phys. **46**(12), 3209–3215 (2007)

Part II

Classical and superposed learning for quantum weightless neural networks

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Classical and superposed learning for quantum weightless neural networks

Adenilton J. da Silva [a,*], Wilson R. de Oliveira [b], Teresa B. Ludermir [a]

[a] Centro de Informática Universidade Federal de Pernambuco, Brazil
[b] Departamento de Estatística e Informática Universidade Federal Rural de Pernambuco, Brazil

ARTICLE INFO

Available online 31 July 2011

Keywords:
Quantum neural networks
Weightless neural networks
Quantum-supervised learning algorithm
Superposition-based learning algorithm

ABSTRACT

A supervised learning algorithm for quantum neural networks (QNN) based on a novel quantum neuron node implemented as a very simple quantum circuit is proposed and investigated. In contrast to the QNN published in the literature, the proposed model can perform both quantum learning and simulate the classical models. This is partly due to the neural model used elsewhere which has weights and non-linear activations functions. Here a quantum *weightless* neural network model is proposed as a quantisation of the classical weightless neural networks (WNN). The theoretical and practical results on WNN can be inherited by these quantum weightless neural networks (qWNN). In the quantum learning algorithm proposed here patterns of the training set are presented concurrently in superposition. This superposition-based learning algorithm (SLA) has computational cost polynomial on the number of patterns in the training set.

1. Introduction

Quantum computation was proposed by Richard Feynman in 1982 [1] motivated by the observation that a quantum system cannot be simulated by a classical computer without an exponential computational cost. In quantum theory of computation, a single quantum computer can follow many distinct computational paths in parallel and produce a final output depending on the interference of all of them. This parallelism enables the proposal of algorithms not matched by classical computation regarding computational costs. Amongst those are the Deutsch-Josza and Simon [2] with an exponential speed up. Shor's Algorithm which solves the factoring problem in polynomial time, a problem believed to be classically intractable. Grover's search algorithm [3] which searches an unordered database quadratically faster then any classical one.

There are several works applying quantum computing in artificial intelligence: quantum neural networks [4–11], decision tree [12], pattern recognition [13] and associative memory [14]. This paper investigates the use of quantum computing techniques to design learning algorithms for neural networks. We propose a quantum weightless neural network and a quantum supervised learning algorithm. The study of quantum weightless neural networks (qWNN) was started by de Oliveira et al. in [4,5] where it is investigated quantisations of probabilistic logic node (PLN) and multi-valued probabilistic logic node (MPLN) [15]. Here we propose and investigate a novel neuron model which is a quantum analogue of the RAM node which we call q-RAM node. The q-RAM node is very simple, but despite its simplicity it can simulate any of its classical siblings, PLN, MPLN, goal seeking neuron (GSN) and pRAM, and their quantisations.

The proposed learning algorithm for training weightless neural networks, the superposition-based learning algorithm (SLA), is based on Grover's search algorithm [3]. The SLA is a quantum supervised learning algorithm for neural networks where all patterns of the training set are presented at the same time to the network using a state in superposition. The computational cost of SLA is polynomial in the number of patterns in the training set. The SLA is able to train any model of weightless neural networks that can be quantised i.e. the free parameters, inputs and outputs of the network can be represented as qubits in different registers and the action of the network can be represented by a quantum operator.

This paper is organised as follows: Sections 2–4 present the basic definitions for quantum computation, classical weightless neural networks and quantum neural networks. Section 5 describes a novel quantum weightless neural node, one of the contributions of this work. Section 6 describes the superposition-based learning algorithm, another contribution of this work. Finally, Section 7 summarises our conclusions and presents future works.

2. Quantum computation

The cross-disciplinary nature of quantum computing makes it difficult to present their main definitions and results unbiased. The presentation which follows is biased towards Mathematics and mostly Computer Science.

The fundamental unit of information in classical computation is the bit which can assume one of the two possible abstract

* Corresponding author.
E-mail address: adenilton.silva@gmail.com (A.J. da Silva).

values in $\mathbb{B}=\{0,1\}$. More complex data types are encoded as sequences of bits. To represent a bit a classical computer must contain a corresponding physical system which can exist in two unambiguously distinguishable states, associated with its two possible abstract values. For example, one such system could be a switch in an open or a shut state; or a magnet whose magnetisation could be in two different orthogonal directions.

Similarly, the fundamental unit of information in quantum computing is the quantum bit or *qubit*. One qubit is a bi-dimensional vector in a (complex) Hilbert space.[1] A qubit represents a state of (bi-dimensional) quantum mechanical system. In actual physical quantum systems, where "Nobody knows *how* it can be like that" [16], Hilbert spaces can be very efficiently used to tell *what* happens.

In this section we briefly introduce the concepts of quantum computing necessary for the remaining sections. For a more complete introduction on Quantum Computation and Information, see [17] or [2] for a more Computing Science oriented approach.

2.1. Quantum bits

The passage from bits to qubits can be understood via mathematical quantisation. A very intuitive view of the quantisation procedure is put forward by Nik Weaver in the preface of his book *Mathematical Quantisation* [18] which briefly says: "The fundamental idea of mathematical quantisation is *sets are replaced with Hilbert spaces*". So the idea is to represent bits 0 and 1 as pairs of orthonormal (column) vectors, a *basis*[2] for $\mathbb{C}^2$. In spite of fact that 0 and 1 can be represented by any orthogonal base of $\mathbb{C}^2$, the mostly used one is the *canonical* (or *computational*) basis defined as the pair

$$|0\rangle=\begin{bmatrix}1\\0\end{bmatrix},\quad |1\rangle=\begin{bmatrix}0\\1\end{bmatrix} \tag{1}$$

where $|0\rangle$ and $|1\rangle$ are, in a notation commonly used by physicists (and quantum computing scientists), the computational basis states and read "ket zero" and "ket one". They are also called basic or ground states. The ket notation was invented in 1939 by Paul Dirac [19] and is related to inner product. This notation is also used for an arbitrary vector $|\psi\rangle$. The other part of the bracket defining the inner product (of say x and y, $\langle x,y\rangle$) is unsurprisingly called *bra*. The bra of a ket vector $|\psi\rangle$ is its conjugate transpose, and thus a row vector, denoted as $\langle\psi|$. Their matrix product $\langle\psi||\psi\rangle$ is a scalar, their inner product. From the inner product we obtain a norm $\||\psi\rangle\|^2=\langle\psi|\psi\rangle$.

2.2. Superposition and parallelism

While in classical computing a one bit state can be only 0 and 1, in quantum computation we can have a continuum of linear combinations of $|0\rangle$ and $|1\rangle$ by the quantisation procedure. For instance the general qubit state $|\psi\rangle=\alpha_0|0\rangle+\alpha_1|1\rangle$ is a state that can be seen as part $|0\rangle$ and part $|1\rangle$ as a *superposition* of the basic states. $|\psi\rangle$ is at the same time in both state $|0\rangle$ and $|1\rangle$.

One of the main properties of quantum computation is the *quantum parallelism*. If one applies a quantum operator $\mathbf{U}_f$ that implements a function $f(x)$ such that $\mathbf{U}_f|x,0\rangle=|x,f(x)\rangle$ in a state in superposition $|\psi\rangle=\sum_{i=0}^{n-1}\alpha_i|x_i,0\rangle$, the value of $f(x)$ will be computed for all qubits $|x_i\rangle$. The resultant state will be $\sum_{i=0}^{n-1}\alpha_i|x_i,f(x_i)\rangle$.

Because of the quantum parallelism, if one have a quantum neural network implemented as a quantum operator, then it will be possible to use states in superposition to evaluate the outputs of all patterns in the training set, all at once in parallel. A drawback is that the individual results of this computation are not direct accessible, due to the properties of the measurement in quantum mechanics.

2.3. Measurement

In Quantum Physics if a system is in a state which is a superposition $|\psi\rangle=\alpha_0|0\rangle+\alpha_1|1\rangle$, upon measurement the system collapses to one of its basis state $|i\rangle, i\in\{0,1\}$ probabilistically:

$$p(|i\rangle)=\frac{|\alpha_i|^2}{\||\psi\rangle\|^2}=\frac{|\alpha_i|^2}{\Sigma_j|\alpha_j|^2}$$

which is the probability that the system will be found in the ground state $|i\rangle$ after a measurement.

After the first measurement of a state $|\psi\rangle$ if one performs another measurements will get the same result. The collapse of the state after measurement says that one cannot see all the results generated by the quantum parallelism. The challenge in quantum computation is how to take advantage of the quantum parallelism before performing a measurement.

2.4. States are rays

Note that if a state is a scalar multiple of another, say $|\phi\rangle=\beta|\psi\rangle$ the chance that the system in state $|\phi\rangle$ will be found, after measurement, in state $|i\rangle, i\in\{0,1\}$ is the same as if the system were in state $|\psi\rangle$

$$p(|i\rangle)=\frac{|\beta\alpha_i|^2}{\||\phi\rangle\|^2}=\frac{\beta^2|\alpha_i|^2}{\beta^2\Sigma_j|\alpha_j|^2}=\frac{|\alpha_i|^2}{\Sigma_j|\alpha_j|^2}$$

and so, the kets $\beta|\psi\rangle$ and $|\psi\rangle$ describe *the same physical system*. The set $\{\alpha|\psi\rangle \mid \alpha\in\mathbb{C}\}$ is the *ray* generated by $|\psi\rangle$ and represents the same state as $|\psi\rangle$. A natural representative of the ray is a normalised vector in the set. As a result, normalising the ket $|\psi\rangle$ i.e. multiplying it by $1/\||\psi\rangle\|$, gives a unit length ket in which the probability of being observed in $|i\rangle$ is

$$p(|i\rangle)=|\alpha_i|^2$$

So the collection of qubits are all the bi-dimensional complex vectors $|\psi\rangle=\alpha_0|0\rangle+\alpha_1|1\rangle$, such that $|\alpha_0|^2+|\alpha_1|^2=1$.

For example, the ket $|\psi\rangle=(1/\sqrt{2})(|0\rangle+|1\rangle)$ represents a system which is $1/2$ equiprobably to be found in any of the two basis states upon measurement.

2.5. Multi-qubits systems

A system with more than one qubit can be represented by the tensor product of their matrix representation. Recall that the tensor product of two bi-dimensional vectors

$$|\psi\rangle=\begin{bmatrix}\psi_0\\\psi_1\end{bmatrix},\quad |\phi\rangle=\begin{bmatrix}\phi_0\\\phi_1\end{bmatrix}$$

is the four-dimensional vector:

$$|\psi\rangle\otimes|\phi\rangle=\begin{bmatrix}\psi_0\begin{bmatrix}\phi_0\\\phi_1\end{bmatrix}\\\psi_1\begin{bmatrix}\phi_0\\\phi_1\end{bmatrix}\end{bmatrix}=\begin{bmatrix}\psi_0\phi_0\\\psi_0\phi_1\\\psi_1\phi_0\\\psi_1\phi_1\end{bmatrix}$$

[1] A complex Hilbert space is a vector space over the complex numbers $\mathbb{C}$ with a *compatible* inner product.

[2] A basis for a vector space $\mathbb{V}$ is any subset of linear independent vectors $B\subseteq\mathbb{V}$ such that any $v\in\mathbb{V}$ is a linear combination of the vectors in B i.e. $v=\Sigma_{x\in B}\alpha_x v$ where $\{\alpha_x\}_{x\in B}$ is a B-indexed set of scalars.

This obviously generalises to any pair of n- and m-dimensional vectors producing a nm-dimensional vector or more generally a (n_0,m_0)-dimensional matrix by a (n_1,m_1)-dimensional one producing a third (n_0m_0,m_0n_0)-dimensional matrix and so on.

Generally we omit the tensor product operator in the notation $|i\rangle \otimes |j\rangle = |i\rangle|j\rangle = |ij\rangle$, where $i,j \in \{0,1\}$. The n-qubits live in the space $\mathbb{C}^{2^n}$.

The construction of $\mathbf{U}_f$ in Section 2.2 for one qubit can be generalised to multi-qubits. Given a Boolean function $f : \mathbb{B}^n \leftrightarrow \mathbb{B}^m$ defines the unitary operator $\mathbf{U}_f : \mathbb{C}^{2^n} \leftrightarrow \mathbb{C}^{2^m}$ where $\mathbf{U}_f|\mathbf{x}\rangle_n|\mathbf{y}\rangle_m = |\mathbf{x}\rangle_n|\mathbf{y} \oplus f(\mathbf{x})\rangle_m$ and $|\mathbf{x}\rangle_n$ are an n qubits base state and $\oplus$ is the bitwise XOR (addition mod 2).

2.6. Linear quantum operators

In Quantum Mechanics observables quantities are *Hermitian operators* whose *eingevectors* form a complete orthonormal basis for the space. The result of a measurement is its *eingenvalue*. The dynamics or time evolution of the system is governed by a *unitary operator* related to the *Halmitonian* of the system. So the next natural step in our quantisation procedure is the representation of the Boolean operators as unitaries.

Four our purposes, a unitary operator $\mathbf{U}$ is a squared matrix over the complex numbers, $\mathbf{U} \in \mathbb{C}^{n\times n}$, such that

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = \mathbf{I}_n$$

where $\mathbf{I}_n$ is the identity $n \times n$ matrix and $\mathbf{U}^\dagger$ is the conjugate transpose (Hermitian conjugate) of $\mathbf{U}$. Being invertible, a unitary operator is reversible. They preserve inner product and so they are isometries. In the Bloch sphere representation of qubits, they correspond to rotations or inversions.

Some examples of operators over one qubit in quantum computation are: **I**, identity operator: does nothing; **X**, flip operator: behaves as the classical NOT on the computational basis and **H**, Hadamard transformation: generates superposition of states. Together with the **X** matrix the **Z** and **Y** matrices (see Eq. (2)) form the well-known Pauli matrices which plays an important role in quantum mechanics in general and in quantum computing in particular. Their matrix representations in relation to the computational basis are displayed in Eqs. (2) and (3).

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2}$$

$$\mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Apart from these gates, the Hadamard gate H, the phase gate S, and the $\pi/8$ -gate T are given by the matrices

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{3}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix}$$

These single qubit gates are important, as they can be used together with the CNOT-gate to give universal sets of discrete quantum gates.

The Hadamard gate can be used to produce equally weighted superpositions as the following simple example shows

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{4}$$

$$\mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{5}$$

The CNOT-gate is an example of a two-qubit controlled operation. It also goes under the name (quantum) XOR. Its matrix representation in the computational basis is

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

The **CNOT** gate performs a **NOT** (i.e. an **X**) operation on the target qubit t conditioned on the control bit c being 1.

2.7. Non-linear quantum operators

A new idea in quantum computing is that the quantum evolution might be slightly non-linear [20]. The non-linearity is useful for overcoming "the difficulties connected with the ordinary linear quantum components" [10]. Following this idea non-linear quantum algorithms have been proposed [21]. The non-linear quantum gates were applied to neural networks in [10,11].

In [20] a non-linear quantum algorithm is proposed that receives a state in superposition $|\psi\rangle = (1/\sqrt{N})\sum_{i=1}^{N}|\psi^i,0\rangle$ and returns the state $|\psi\rangle = (1/\sqrt{N})\sum_{i=1}^{N}|\psi^i,0\rangle$ or $|\psi\rangle = (1/\sqrt{N})\sum_{i=1}^{N}|\psi^i,1\rangle$. The last qubit is changed to $|1\rangle$ if and only if in the superposition all ψ_i are equal to 0. Note that his operator could be used to solve the satisfiability problem.

2.8. Quantum circuits

Quantum operators can be represented as quantum circuits. Fig. 1 shows the quantum circuit for the **CNOT** gate and Fig. 2 shows a quantum circuit where an n-qubit controlled gate U whose action on the target qubit (bottommost) is active or not by $n-1$ (topmost) control qubits [17]. The output is checked by measurement gates.

2.9. Grover's algorithm

Grover's algorithm is a quantum algorithm for searching an unordered data quadratically faster than any classical method [22]. Given a data set with N items the most efficient classical algorithm will need execute an average of 0.5 N classical steps before finding the desired item. In the worst case, N classical steps are necessary. Grover's algorithm outperforms any classical algorithm and realises the task with $O(\sqrt{N})$ quantum steps. The iteration number T of the algorithm is very important and is calculated as in Eq. (7), where M is the number of answers in the search space:

$$T = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor \tag{7}$$

The Algorithm 1 is based on the one shown in [2] and M is a squared 2^n-dimensional matrix whose each entry is $1/2^n$.

Algorithm 1 (*Grover's algorithm*).

1 Initialise the system $|\psi\rangle = |0\rangle_n$
2 Apply the Hadamard Transform $|\psi\rangle = H^{\otimes n}|\psi\rangle$
3 Apply the phase inversion operation: $U_f(\mathrm{I} \otimes \mathrm{H})$
4 Apply the inversion about the mean operation: $-\mathrm{I}+2\mathrm{M}$
5 Repeat steps 3 and 4, $T = O(\sqrt{2^n})$ times.
6 Measure the state $|\psi\rangle$

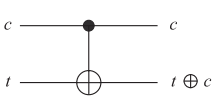


Fig. 1. Controlled NOT gate.

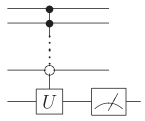


Fig. 2. A quantum circuit for the general controlled **U** gate where **U** is an arbitrary 1-qubit unitary operator. Note that appropriately choosing circles (○) and bullets (•) one could define a controlled operator which would apply **U** to the bottommost qubit if the corresponding binary combination (0′ s for circles and 1′ s for bullets) is present in the remaining qubits from top to bottom.

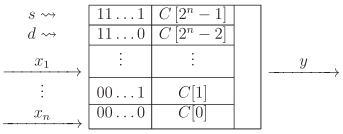


Fig. 3. RAM node.

The iterations 3 and 4 of Algorithm 1 can be represented as a quantum operator **G** called the Grover iteration [22].

3. Weightless neural networks

The RAM-based neural networks were proposed by Igor Aleksander [23] and do not have weights associated in their connections.

A RAM node with n inputs has 2^n memory locations, addressed by the n-bit string $a=(a_1a_2\ldots a_n)$. A binary signal $x=(x_1x_2\ldots x_n)$ on the input lines will access only one of these locations resulting in $y=C[x]$ [15]. In Fig. 3, s and d are respectively the learning strategy and the desired output to be learned.

Learning in weightless neural networks takes place simply by writing into the corresponding look-up table entries. This learning process is much simpler than the adjustment of weights. In spite of the simplicity of the RAM-based nodes, RAM-based networks have good generalisation capabilities [24] and computational power [25].

The PLN is based on the RAM node. The difference between the PLN and RAM nodes is that a 2-bit number (rather than a single bit) is now stored at the addressed memory location. The content of this location is turned into the probability of firing (i.e. generating 1) at the overall output of the node. In other words, a PLN consists of a RAM node, where now a 2-bit number 00, 11 and 01 (or 10) stored at the addressed memory location are to be interpreted respectively as 0, 1 or u. The output of the PLN Node is given by [15]

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

The multi-valued probabilistic logic node (MPLN) differs from PLN by allowing a wider but still finite set of finite precision probabilities $M=\{p_0,\ldots,p_{2^n-1}\}$ to be stored at each memory content. The output of the MPLN Node is given by [15]

$$y=\begin{cases}0 & \text{if} \quad C[x]=0\\ 1 \quad \text{withprobability}\ \dfrac{1}{p} & \text{if} \quad C[x]=p\end{cases}$$

The goal seeking neuron (GSN) differs from the PLN by allowing that the node receives and generate values 0, 1 and u. If the GSN node receives one value u in its input lines a set of memory positions will be addressed. The set of addressed positions is derived from the original address replacing the u values in the input vector to 0 and 1. For instance, if a three input GSN receives the input $I=0u1$, two addresses of the GSN node will be accessed $a_1=001$ and $a_2=011$ because of the u value in the second position of I. The output of the GSN node is determined by the number of values 0, 1 and u in the addressed memory positions.

4. Quantum weighted neural networks

The concept of quantum neural computation was first introduced by Kak in 1995, creating a new paradigm using neural networks and quantum computation which opened new directions in neural network research [26]. It is expected that quantum neural networks are more efficient than classical neural networks, parallel to what is expected from quantum computation in relation to classical computation.

Since the Kak's work further neural networks models have been proposed [6,7,10,4,9,27,28], but there remains the challenge of direct implementation in quantum circuits, natural adaptation of the classical learning algorithms and quantum learning algorithms respecting the postulates of the quantum mechanics. These are characteristics not altogether found in any of the proposed quantum weighted neural networks models but are found in our new model.

In this section we describe some models of quantum neural networks and their learning algorithms. We verify that the learning algorithms for these models break the postulates of quantum computing by the use of non-linear or non-unitary quantum operators. Use of non-linear quantum operators is an open question, but non-linear quantum computation implies P=NP [20].

The proposed learning algorithms for quantum neural networks [6,10,28,9] can be classified as iterative [28,9] (in each iteration only one pattern is presented to the network) or superposition based [10,6] (all patterns are presented to the networks concurrently in superposition). Here we analyse one iterative algorithm [28] and one based on the superposition [10] and we show that these algorithms for weighted neural networks are non-unitary and non-linear.

The one qubit output $|y\rangle$ of a quantum perceptron [28] with inputs $|x_1\rangle,\ldots,|x_n\rangle$ is defined in Eq. (8) where $\hat{w}_j$ are 2×2 matrices representing the weights of neuron and $\hat{F}$ is a quantum operator. The quantum iterative learning rule for this model is presented with $F=I$ in Eq. (9), where t is the iteration number, $|d\rangle$ is the desired output and the output of the node in time t is described by Eq. (10):

$$|y\rangle=\hat{F}\sum_{j=1}^{n}\hat{w}_j|x_j\rangle \tag{8}$$

$$|y(t)\rangle=\sum_{j=1}^{n}\hat{w}_j(t)|x_j\rangle \tag{9}$$

$$\hat{w}_j(t+1)=\hat{w}_j(t)+\eta(|d\rangle-|y(t)\rangle)\langle x_j| \tag{10}$$

The learning rule of quantum perceptron drives the quantum perceptron into the desired output $|d\rangle$ [28], but the learning rule (10) does not preserve unitary operators.

Theorem 4.1. *The learning rule described in Eq.* (10) *does not preserve unitary operators.*

Proof. We construct a counterexample and execute one iteration of the iterative quantum learning rule. In Eq. (10) set $j=1$, the weight $\hat{w}_1(t)=I$, the desired output $|d\rangle=|1\rangle$, the network output is $|0\rangle$, the input of $\hat{w}_1(t)$ is $|x_j\rangle=|1\rangle$ and $\eta=0.5$. Then

$$\hat{w}_j(t+1)=I+0{,}5(|1\rangle-|0\rangle)\langle 1|=I+0{,}5(|1\rangle\langle 1|-|0\rangle\langle 1|)$$
$$=\begin{pmatrix}1&0\\0&1\end{pmatrix}+\begin{pmatrix}0&-0.5\\0&0.5\end{pmatrix}=\begin{pmatrix}1&-0.5\\0&1.5\end{pmatrix} \quad (11)$$

where we see that $w_1(t+1)$ is non-unitary. □

One may think that the trouble lies in the choice of the learning rate but let η be arbitrary. Then

$$\hat{w}_1(t+1)=\begin{pmatrix}1&-\eta\\0&1+\eta\end{pmatrix}\rightarrow\hat{w}_1(t+1)\hat{w}_1(t+1)^{\dagger}$$
$$=\begin{pmatrix}1+\eta^2&-\eta-\eta^2\\-\eta-\eta^2&(1+\eta)^2\end{pmatrix} \quad (12)$$

and since $\hat{w}_1(t+1)$ is unitary

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

and we must have $\eta=0$. In this case there is no non-zero learning rate that allows the use of rule (10) without violating the quantum computing postulates.

The learning algorithms that use states in superposition [10,6] also violates the quantum computation postulates. In [10,11] it is used as a non-linear quantum operator proposed in [20] and in [6] it is necessary for the use of a non-linear quantum oracle. One difficulty here is that non-linear quantum mechanics implies $P=NP$ [20].

5. Quantum weightless neural networks

In [6] we define a quantisation of the RAM node, the qRAM node. Quantum weightless neural networks do not use non-linear activation function, like sigmoid or tangent hyperbolic. This is important because non-linear activation functions will hardly have an exact quantum analogous [28].

Usually the RAM node stores one bit at each addressable location. We follow this approach by investigating first the qRAM node which store just one qubit. This qubit will not be directly stored in a quantum register as in the classical case. We will rather use a selector (parameter) and an operator which applied to the selector will produce the desired qubit. This form of quantisation of weightless neural networks was proposed in [4,5] using different matrices from the matrix **A** used here

$$\mathbf{A}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix} \quad (13)$$

The matrix **A** defines a quantum operator over two qubits that simply flip the second qubit if the first is in the state $|1\rangle$ and it does nothing if the first state is in the state $|0\rangle$. The reader familiar with quantum computing terminology will notice that our matrix **A** is well known as the *controlled not* or c-NOT gate. We keep with the alternative notation since it gives rooms for further generalisations as in [4,5] where more complex matrices are used.

Definition 5.1. A qRAM node with n inputs is represented by the operator **N** described in Eq. (14). The inputs, selectors and outputs of **N** are organised in three quantum registers $|i\rangle$ with n qubits, $|s\rangle$ with 2^n qubits and $|o\rangle$ with 1 qubit. The quantum state $|i\rangle$ describes qRAM input, and quantum state $|s\rangle|o\rangle$ describes qRAM state.

$$\mathbf{N}=\sum_{i=0}^{2^n-1}|i\rangle_n\langle i|_n A_{s_i,o} \quad (14)$$

Fig. 4 describes the quantum circuit of a qRAM node with two inputs $|\psi\rangle$ and $|\varphi\rangle$. Four selectors $|s_i\rangle$ and four operators $\mathbf{A}_{s_i,o}$ each one equals to the **A** operator above, where the first qubit is the selector $|s_i\rangle$ and the second is the state in the output register $|o\rangle$. We can now see that learning is achieved by adapting the value of the selectors to the training set. The main advantage of qRAM over classical weightless neural networks is its capacity to receive input and selectors in superposition. When the selectors are in the computational basis the qRAM node acts exactly as a RAM node. But it behaves in a much more interesting way when is fed with a superposition of basis state.

Let us see the operation of qRAM node **N** when given a value in the form $|\psi\rangle=(1/\sqrt{2})|0\rangle+(1/\sqrt{2})|1\rangle$ in one of its input lines. Further assume that $|\varphi\rangle=0$, $|s_1s_2s_3s_4\rangle=|0111\rangle$ and $|o\rangle=0$. Eq. (15) shows the operation of the qRAM. In this example the two matrices A_0 and A_2 are simultaneously addressed and the output register is in state $|o\rangle=(1/\sqrt{2})(|0\rangle+|1\rangle)$ i.e. outputs of 00 and 10 are calculated simultaneously.

$$\mathbf{N}\left[\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle|0111\rangle|0\rangle\right]$$
$$=\frac{1}{\sqrt{2}}[\mathbf{N}(|0\rangle|0\rangle|0111\rangle|0\rangle)+\mathbf{N}(|1\rangle|0\rangle|0111\rangle|0\rangle)]$$
$$=\frac{1}{\sqrt{2}}[(|0\rangle|0\rangle|0111\rangle|0\rangle)+(|1\rangle|0\rangle|0111\rangle|1\rangle)] \quad (15)$$

One example of qRAM network is presented in Fig. 5, qRAM networks have pyramidal architecture and low connectivity. The configuration of a given qRAM network is represented by a string of qubits $(s_{11},s_{12},\ldots,s_{1n};\ldots;s_{m1},s_{m1},s_{m2}\ldots,s_{mn})$ representing network selectors, where m is the number of neurons and n is the number of inputs of neurons.

$$\mathbf{N}=|00\rangle\langle 00|\otimes A_{s_1,o}+|01\rangle\langle 01|\otimes A_{s_2,o}$$
$$+|10\rangle\langle 10|\otimes A_{s_3,o}+|11\rangle\langle 11|\otimes A_{s_4,o} \quad (16)$$

Exponential cost in simulation of quantum computers makes it prohibitive to simulate qRAM networks. Here we present a simple example to show qRAM network operation. We can use the qRAM network in Fig. 5 to solve the four bit parity problem. The

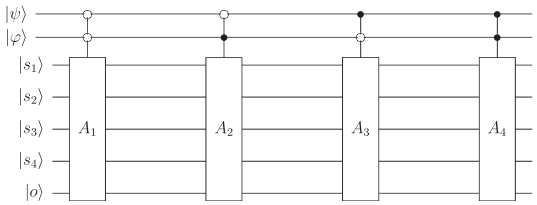


Fig. 4. qRAM node.

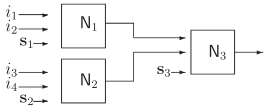


Fig. 5. Two-layer qRAM network.

configuration of the network is $\mathbf{s}_1 = |0110\rangle$, $\mathbf{s}_2 = |0110\rangle$ and $\mathbf{s}_3 = |0110\rangle$. One can present inputs simultaneously as in Eq. (17) to qRAM network

$$|i\rangle = \frac{1}{2}(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle) \tag{17}$$

The neuron $\mathbf{N}_1$ receives as input the two first qubits of state $|i\rangle$, $|i_1 i_2\rangle = |00\rangle$. The action of neurons of network is described by the operator in Eq. (16). The action of the neuron $\mathbf{N}_1$ and $\mathbf{N}_2$ are described in Eqs. (18) and (19)

$$\begin{aligned}\mathbf{N}_1|i_1 i_2\rangle|\mathbf{s}_1\rangle|o_1\rangle &= (|00\rangle\langle 00| \otimes A_{s_1,o} + |01\rangle\langle 01| \otimes A_{s_2,o} \\ &+ |10\rangle\langle 10| \otimes A_{s_3,o} + |11\rangle\langle 11| \otimes A_{s_4,o})|00\rangle|0110\rangle|0\rangle \\ &= |00\rangle\langle 00| \otimes A_{s_1,o}(|00\rangle|0110\rangle|0\rangle) = |00\rangle|0110\rangle|0\rangle\end{aligned} \tag{18}$$

$$\begin{aligned}\mathbf{N}_2|i_3 i_4\rangle|\mathbf{s}_2\rangle|o_2\rangle &= (|00\rangle\langle 00| \otimes A_{s_1,o} + |01\rangle\langle 01| \otimes A_{s_2,o} \\ &+ |10\rangle\langle 10| \otimes A_{s_3,o} + |11\rangle\langle 11| \otimes A_{s_4,o})\frac{1}{2}(|00\rangle \\ &+ |01\rangle + |10\rangle + |11\rangle)|00\rangle|0110\rangle|0\rangle \\ &= \frac{1}{2}(|00\rangle\langle 00| \otimes A_{s_1,o}(|00\rangle|0110\rangle|0\rangle) \\ &+ |01\rangle\langle 01| \otimes A_{s_2,o}(|00\rangle|0110\rangle|0\rangle) \\ &+ |10\rangle\langle 10| \otimes A_{s_3,o}(|00\rangle|0110\rangle|0\rangle) \\ &+ |11\rangle\langle 11| \otimes A_{s_4,o}(|00\rangle|0110\rangle|0\rangle)) = |00\rangle|0110\rangle|0\rangle \\ &+ |01\rangle|0110\rangle|1\rangle + |10\rangle|0110\rangle|1\rangle + |11\rangle|0110\rangle|0\rangle\end{aligned} \tag{19}$$

The outputs of neurons $\mathbf{N}_1$ and $\mathbf{N}_2$ are $|o_1\rangle = |0\rangle$ and $|o_2\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle)$. These outputs will be used as inputs of neuron $\mathbf{N}_3$. Eq. (20) shows the action of $\mathbf{N}_3$ and the network calculates the outputs of all the inputs in superposition

$$\begin{aligned}\mathbf{N}_3|o_1 o_2\rangle|\mathbf{s}_3\rangle|o_3\rangle &= \frac{1}{\sqrt{2}}(|00\rangle\langle 00|A_{s_1 o_3}|00\rangle|0110\rangle|0\rangle \\ &+ |01\rangle\langle 01|A_{s_2 o_3}|01\rangle|0110\rangle|0\rangle) \\ &= \frac{1}{\sqrt{2}}(|00\rangle|0110\rangle|0\rangle + |01\rangle|0110\rangle|1\rangle)\end{aligned} \tag{20}$$

The action of the qRAM network **Net** can be summarised as in Eq. (21). The network calculates the output of each input in superposition

$$\begin{aligned}&\mathbf{Net}\left(\frac{1}{2}(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle)\right)|0\rangle \\ &= \frac{1}{2}(\mathbf{Net}|0000\rangle|0\rangle + \mathbf{Net}|0001\rangle|0\rangle \\ &+ \mathbf{Net}|0010\rangle|0\rangle + \mathbf{Net}|0011\rangle|0\rangle) \\ &= \frac{1}{2}(|0000\rangle|0\rangle + |0001\rangle|1\rangle + |0010\rangle|1\rangle + |0011\rangle|0\rangle)\end{aligned} \tag{21}$$

In Section 6 the capacity to process states in superposition will be explored to train a neural network.

5.1. Simulation of classical weightless models

Interestingly, if we perform a measurement at the output wire of the qRAM and allow for superposition of the selectors the qRAM node can simulate any of its siblings PLN, MPLN and pRAM and the training algorithm of the classical weightless neural networks can be adapted to the qRAM.

Let $|u\rangle = H|0\rangle$ be an undefined state as in Section 3, in Eq. (15) the qRAM node behaves as a GSN node. The node can receive and produce $|0\rangle$, $|1\rangle$ and $|u\rangle$ then the qRAM node can be viewed as a qGSN node, but the qRAM node can receive and produce others quantum states behaving as a sort of *continuum* valued qGSN node.

Algorithm 2 (*PLN net learning*).

```
1  All memory contents are set to u;
2  while some stopping criterion is not met do
3      One of the N training patterns p
       is presented to the net;
4      learn ← FALSE
5      for t = 1 to η do
6          The net is allowed to produce the output for the
           pattern p
7          if s is equal to the desired output for the
           pattern p then
8              learn ← TRUE
9              break
10         end
11     end
12     if learn then
13         all the addressed memory content are made 7
             to assume their
           current output values, making those with u
            become definitely 0
           or 1, accordingly
14     else
15         all the addressed memory content are made to assume
           the u value
16     end
17 end
```

The classical algorithm of PLN network is defined in Algorithm 2. In order to simulated a PLN the values stored in a PLN node 0, 1 and u, will be represented respectively by the qubits $|0\rangle$, $|1\rangle$ and $|u\rangle = H|0\rangle$. The probabilistic output of the PLN is obtained with a measurement in the output of the qRAM node. The results of this measurement is described in Eq. (22)

$$y = \begin{cases} 0 & \text{if} \quad |o\rangle = |0\rangle \\ 1 & \text{if} \quad |o\rangle = |1\rangle \\ \mathit{random}(0,1) & \text{if} \quad |o\rangle = |u\rangle \end{cases} \tag{22}$$

With a small modification of Algorithm 2 one can train the qRAM node. It is necessary to measure the output of each node and define $|u\rangle = \mathbf{H}|0\rangle$, replace the line 1 of Algorithm 2 with *all selectors are set to* $|u\rangle$ and replace lines 13 and 15 respectively with *all the addressed A gates have their selectors changed to produce the current output of their nodes, making those with* $|u\rangle$ *become definitely* $|0\rangle$ *or* $|1\rangle$*, accordingly* and *all the addressed A gates have their selectors set to* $|u\rangle$. We can see the PLN node as a particular case of the qRAM where a measurement is realised in the output of each node i.e. we are working only with classical data.

The Fig. 6 shows the quantisation of lines 5–11 in Algorithm 2. Operator *Net* represents a qRAM network with input $|p\rangle$, selectors $|s\rangle$ and output $|o\rangle$. There are η *Net* operators to perform

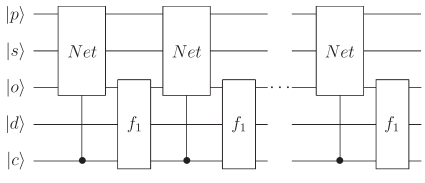


Fig. 6. PLN learning iteration.

the for loop in Algorithm 2. The operator f_1 change the control register $|c\rangle$ to $|0\rangle$ if the desired output $|d\rangle$ is equal to network output $|o\rangle$ and stop the execution of the networks. Operator f_1 also changes $|o\rangle$ to $|0\rangle$ if $|d\rangle$ is different from $|o\rangle$ preparing the output register to the next *Net* operator.

Algorithm 3 shows the quantisation of Algorithm 2. In this algorithm all neuron outputs are measured, data in registers $|p\rangle$, $|o\rangle$, $|d\rangle$ and $|c\rangle$ are in computational basis. Only the register $|s\rangle$ receives states in superposition.

Algorithm 3 (*Naive qRAM net learning*).

1 All selectors $|s\rangle$ are set to $|u\rangle$;
2 **while** *some stopping criterion is not met* **do**
3 One of the N training patterns $|p\rangle$ is presented to the net;
4 Set $|c\rangle=|1\rangle$ and $|o\rangle=0$
5 Let $d(p)$ be the desired output of pattern p. Set $|d\rangle=|d(p)\rangle$
6 Apply the quantum circuit described in Fig.6
7 **if** $|c\rangle=|0\rangle$
8 all the addressed memory content are made to assume their current output values, making those with $|u\rangle$ become definitely $|0\rangle$ or $|1\rangle$, accordingly
9 **else**
10 all the addressed memory content are made to assume the $|u\rangle$ value
11 **end**
12 **end**

In [4,5] the size of matrices depends on the number of stored probabilities which make the quantisation of the pRAMs impossible. In contrast our novel model has constant size matrices independent of the number of stored probabilities. The probability is stored as amplitude of the input state and/or the selectors which can be tuned during the learning phase through phase rotations.

One can easily simulate the other weightless neural nodes with the qRAM node. Then any computation done by a weightless neural node can be realised by a qRAM. This implies, for instance, that the qRAM maintain the generalisations capabilities of the weightless models. In the next section we present a quantum learning algorithm to train the qRAM node.

Algorithm 3 is called naive because it does not explore quantum mechanics properties. In next section we present a quantum algorithm to train qRAM networks exploring its capacity to receive states in superposition.

6. Superposition-based learning algorithm

Let us imagine a classically improbable learning algorithm. For a fixed architecture the number of possible WNN is finite, say n. Supposing we have all the required computational resource available, we could present all the p patterns in the learning set at once in parallel to each individual network; perhaps making p copies of each one of the n networks. Then we could search amongst the pn combinations networks and input patterns that network which recognises the training set mostly accordingly to some fixed criterion. The quantisation of this absurdly computational expensive algorithm is what we propose next and show that in a quantum computer it is a polynomial time algorithm!

The superposition-based learning algorithm (SLA) is a supervised algorithm that takes advantage of states in superposition to approximate the idea in the previous paragraph. SLA can be used to train qRAM networks where the selectors and inputs are a quantum state $|\psi\rangle$ composed of four quantum registers. The first register **s** will store the selectors, the second register **p** will store the training pattern, the third register **o**, with one qubit, will store the output of the node and the fourth register **d**, used only in the learning phase of the algorithm, will store the desired output.

Algorithm 4 (*Superposition-based learning*).

1 Initialise all the qubits in register **s** with the quantum state $\mathbf{H}|0\rangle$.
2 Initialise the register **p**, **o**, **d** with the quantum state $|p\rangle=\sum_{i=1}^{p}|p_i,0,d_i\rangle$.
3 $|\psi\rangle=\mathbf{N}|\psi\rangle$, where **N** is a quantum operator representing the action of the neuron.
4 Use a quantum oracle to change the phase of the states where the registers **p**, **o**, **d** $=\sum_{i=1}^{p}|p_i,d_i,d_i\rangle$
5 Apply the operator inverse to the neuron in the state $|\psi\rangle$, $|\psi\rangle=\mathbf{N}^{-1}|\psi\rangle$, to disentangle the state in the register s
6 Apply the inversion about the mean operation (see Algorithm 1) in the register **s**
7 Repeat steps 3–6, $T=(\pi/4)\sqrt{n}$ times, where n is the number of selectors of the networks.
8 Measure the register **s** to obtain the desired parameters.

In step 1 of Algorithm 4 a superposition of all possible configurations for the network is created. This step is realised setting the quantum register **s** to $|0,\ldots,0\rangle$ and applying the Hadamard operator in all qubits of the register **s**, at this moment **s** will hold the quantum state $|s\rangle$ described in Eq. (23), where m is the number of selectors, $a=2^m-1$ and $s_j=j$

$$|s\rangle=\frac{1}{\sqrt{a+1}}\sum_{j=0}^{a}|s_j\rangle \tag{23}$$

Let p_i be a pattern of the training set and d_i be its desired output. In step 2 a superposed quantum state holding all p_i and d_i is set to the quantum registers **p** and **d** respectively. The output register **o** is set to the basis state $|0\rangle$.

One can execute step 2 as in the case of quantum associative memory proposed in [14,29]. At this moment the register **p** holds all training patterns p_i and the register **d** holds all desired outputs d_i. The state of the registers **p**, **o** and **d** is described in Eq. (24)

$$|\mathbf{p},\mathbf{o},\mathbf{d}\rangle=\frac{1}{\sqrt{p}}\sum_{i=0}^{p-1}|p_i,0,d_i\rangle \tag{24}$$

Let **N** be the operator associated to a quantum neural network. The action of **N** may be summarised as sending the state $|s,p_i,0,d_i\rangle$ into the state $|s,p_i,y_i,d_i\rangle$, where y_i is the calculated output of the node.

In step 3 of Algorithm 4 the operator **N** acts on the state $|\psi_0\rangle$ and, by linearity of the quantum operators, the desired output is produced in all terms of the superposition. Eq. (25) describes the state $|\psi_1\rangle$, where m is the cardinality of the training set and $a=2^m-1$

$$|\psi\rangle=\frac{1}{\sqrt{a+1}}\frac{1}{\sqrt{p}}\sum_{i=0}^{p-1}\sum_{j=0}^{a}|s_j,p_i,y_{ij},d_i\rangle \tag{25}$$

Let **O** be the quantum oracle that sends the state $\sum_{i=0}^{p-1}|p_i,d_i,d_i\rangle$ into the state $-\sum_{i=0}^{p-1}|p_i,d_i,d_i\rangle$ and that acts as the identity for the other states. In step 4 of the SLA the oracle **O** receives the state $|\psi_1\rangle$ and will mark only the states where the desired outputs are equals to

the calculated output $y_{ij}=d_i$. The action of the oracle will permit to use the Grover's search algorithm in the register **s**.

In the third step the neuron **N** entangled the quantum register **s** with the quantum registers, **p**, **o** and **d**, then the quantum search algorithm applied in the register s will change the registers **p**, **o** and **d**. To avoid this change in the registers **p**, **o** and **d** the operator $\mathbf{N}^{-1}$ is applied to disentangle the states in the registers in the fifth step of the algorithm. Then the inversion about the mean operation is applied in the register **s**. The steps 3–6 are repeated T times and a measurement will return the desired parameters.

6.1. Complexity of SLA

In this section we analyse the complexity of the SLA algorithm. To simplify the analysis we suppose that all possible (Boolean) patterns are in the training set, the number of patterns in training set is $N_p=2^n$ and each q-RAM node has two inputs. In this situation patterns can be represented with $n=\log N_p$ bits. A pyramidal qRAM network with $\log N_p$ input terminals will have $2^{\log(\log N_p)}-1$ neurons. Then the number of qubits in the quantum register **s** will be $4(\log N_p-1)$. The search occurs in the quantum register s with all $2^{4(\log N_p-1)}$ possible selectors and this search will have cost $\pi/4\sqrt{2^{4(\log N_p-1)}}=(\pi/4)(2^{2(\log N_p-1)})=(\pi/4)(2^{-2}\cdot N_p^2)=(\pi/16)(N_p^2)$. Then the SLA algorithm has polynomial time in the number of patterns N_p in the training set. This result is similar to the cost of the learning algorithm proposed in [10].

6.2. Training a q-RAM node with the SLA

Let us look at the concrete example of the SLA training the q-RAM node to solve the XOR problem. We shall see that a q-RAM node with two inputs learn the function with only one iteration. The training is set to $S=\{(00,0),(01,1),(10,1),(11,1)\}$.

A q-RAM node with two inputs has four selectors, then in step 1 the state $|s\rangle$ is initialised as

$$|s\rangle=H^{\otimes 4}|0000\rangle=\frac{1}{4}\sum_{i=0}^{16}|i\rangle_4 \tag{26}$$

In step 2 the registers **p**, **o** and **d** are prepared as in Eq. (27)

$$|p,o,d\rangle=\frac{1}{2}(|00,0,0\rangle+|01,0,1\rangle+|10,0,1\rangle+|11,0,1\rangle) \tag{27}$$

At this moment the state $|\psi\rangle$ can be described as in Eq. (28)

$$|\psi\rangle=\frac{1}{8}\sum_{i=0}^{16}|i\rangle_4(|00,0,0\rangle+|01,0,1\rangle+|10,0,1\rangle+|11,0,1\rangle) \tag{28}$$

In step 3 the operator **N** is applied to $|\psi\rangle$ which calculates the output y_p^i of the node for each selector i and for each pattern p as in Eq. (29)

$$|\psi\rangle=\frac{1}{8}\sum_{i=0}^{16}|i\rangle_4(|00,y_{00}^i,0\rangle+|01,y_{01}^i,1\rangle+|10,y_{10}^i,1\rangle+|11,y_{11}^i,1\rangle) \tag{29}$$

In step 4 an oracle **O** which inverts the phase of state $|e_1\rangle=(|00,0,0\rangle+|01,1,1\rangle+|10,1,1\rangle+|11,0,0\rangle)$ is applied to $|\psi\rangle$ and we obtain the state in Eq. (30), where δ is the Kronecker's delta and $x(i)=1$ if and only if $|p_i,o_i,d_i\rangle=|e_1\rangle$. The oracle marks the state with the desired parameters

$$|\psi\rangle=\frac{1}{8}\sum_{i=0}^{16}(-1)^{\delta_{1x(i)}}|i\rangle_4(|00,y_{00}^i,0\rangle+|01,y_{01}^i,1\rangle+|10,y_{10}^i,1\rangle+|11,y_{11}^i,1\rangle) \tag{30}$$

In step 5 the operators $\mathbf{N}^{-1}$ and **G** are applied to $|\psi\rangle$ and the amplitude of the marked state goes to one and the amplitude of the others state go to zero. The training finishes in three iterations because $T=\lfloor(\pi/4)\sqrt{16}\rfloor=3$, then one can measure the **s** register and use the result as the parameters of the network.

7. Conclusion

A quantum weightless neural node based on the quantisation of the RAM node, the qRAM node, was proved. Its ability to simulate the classical weightless neural networks was demonstrated, a very important result since all the theoretical and practical results for WNN are inherited by our model. The main property of the qRAM node explored in this paper is the ability to receive all patterns from the training set in superposition.

The size of the matrices A in other models of quantum weightless neural networks is exponential on the number of stored probabilities [4,5]. Our model can store probabilities using matrices A with constant size. The probabilities are stored in the amplitudes of the selectors not in the matrices A. This reduction in space complexity may allow classical simulations of networks composed by qRAM nodes for small sized problems.

We also propose a quantum learning algorithm for neural networks, the superposition-based learning algorithm (SLA). The SLA is a supervised learning algorithm. It explores the capacity of the qRAM to receive qubits in superposition and apply a retrieval algorithm of a quantum probabilistic memory to choose the best configuration of the network. The SLA receives several neural network configurations in superposition and returns a trained neural network.

The SLA has polynomial computational cost in the number of patterns in the training set. This cost is mainly associated with the use of the quantum search algorithm. Because of the superposition capacity one can argue that it may be possible to design an algorithm with a smaller computational cost exploring superposition of neural networks.

A possible future work is to create an ensemble of neural networks in a single quantum neural network with a polynomial cost in the size of the training set. This task may be realised with an algorithm that marks a parameter (one qubit) of the networks in superposition. Instead of performing a measurement projecting to the basis state one can realise a projective measurement to collapse the network to the marked networks.

Another possible future work is to develop a probabilistic version of the SLA where we must run the neural network only twice, one forward and other backward. This can be obtained changing the Grover algorithm for a retrieval algorithm of a quantum probabilistic memory as in [13].

Acknowledgements

This work is supported by research grants from CNPq, CAPES and FACEPE (Brazilian research agencies).

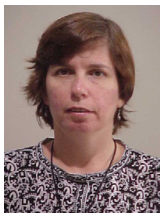
References

[1] R. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (1982) 467–488.
[2] N.S. Yanofsky, M.A. Mannucci, Quantum Computing for Computer Scientists, Cambridge University Press, New York, NY, USA, 2008.
[3] L.K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, ACM, 1996, pp. 212–219.
[4] W. de Oliveira, A.J. Silva, T.B. Ludermir, A. Leonel, W. Galindo, J. Pereira, Quantum logical neural networks, in: Neural Networks, 2008. SBRN '08. 10th Brazilian Symposium on 2008, pp. 147–152.

[5] W. de Oliveira, Quantum RAM based neural networks, in: European Symposium on Artificial Neural Networks 2009. Proceedings 2009, pp. 331–336.
[6] A.J. Silva, W. de Oliveira, T.B. Ludermir, A weightless neural node based on a probabilistic quantum memory, in: Neural Networks, 2010. SBRN '10. 11th Brazilian Symposium on 2010, pp. 259–264.
[7] A.J. Silva, W. de Oliveira, T.B. Ludermir, Superposition based learning algorithm, in: Neural Networks, 2010. SBRN '10. 11th Brazilian Symposium on 2010, pp. 1–6.
[8] 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, Cambridge, MA, 2004, pp. 1–6.
[9] R. Zhou, Q. Ding, Quantum M-P neural network, International Journal of Theoretical Physics 46 (2007) 3209–3215.
[10] M. Panella, G. Martinelli, Neural networks with quantum architecture and quantum learning, International Journal of Circuit Theory and Applications 39 (1) (2011) 61–77, doi:10.1002/cta.619.
[11] M. Panella, G. Martinelli, Neurofuzzy networks with nonlinear quantum learning, IEEE Transactions on Fuzzy Systems 17 (3) (2009) 698–710.
[12] E. Farhi, S. Gutmann, Quantum computation and decision trees, Physical Review A 58 (2) (1998) 915–928.
[13] C.A. Trugenberger, Quantum pattern recognition, Quantum Information Processing 1 (2002) 471–493.
[14] D. Ventura, T. Martinez, Quantum associative memory, Information Sciences 124 (1–4) (2000) 273–296.
[15] T.B. Ludermir, A. Carvalho, A.P. Braga, M.C.P. Souto, Weightless neural models: a review of current and past works, Neural Computing Surveys 2 (1999) 41–61.
[16] R. Feynman, The Character of Physical Law, The MIT Press, 2001.
[17] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.
[18] N. Weaver, Mathematical Quantization, Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, FL, 2001.
[19] P. Dirac, The Principles of Quantum Mechanics, Clarendon Press, 2004.
[20] D.S. Abrams, S. Lloyd, Nonlinear quantum mechanics implies polynomial-time solution for *NP*-complete and *#P* problems, Physical Review Letters 81 (18) (1998) 3992–3995.
[21] M. Czachor, Notes on Nonlinear Quantum Algorithms, ArXiv:quant-ph/9802051v2.
[22] L.K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Physical Review Letters 79 (2) (1997) 325–328.
[23] I. Aleksander, Self-adaptive universal logic circuits, Electronics Letters 2 (8) (1966) 321–322.
[24] A.F. Souza, F. Pedroni, E. Oliveira, P.M. Ciarelli, W.F. Henrique, L. Veronese, C. Badue, Automated multi-label text categorization with VG-RAM weightless neural networks, Neurocomputing 72 (10–12) (2009) 2209–2217.
[25] M. de Souto, T. Ludermir, W. de Oliveira, Equivalence between RAM-based neural networks and probabilistic automata, IEEE Transactions on Neural Networks 16 (4) (2005) 996–999.
[26] S.C. Kak, On quantum neural computing, Information Sciences 83 (3) (1995) 143–160.
[27] Z. Righi, J. Nan, D. Qiulin, Model and training of QNN with weight, Neural Processing Letters 24 (3) (2006) 261–269.
[28] M.V. Altaisky, Quantum Neural Network, Type Technical Report, Institution Joint Institute for Nuclear Research, Russia, available online at the Quantum Physics repository arXiv:quant-ph/0107012v2, 2001.
[29] C. Trugenberger, Probabilistic quantum memories, Phys. Rev. Lett. 87 (6) (2001) 067901.

Adenilton da Silva received the Licentiate degree in mathematics and M.Sc degree in Computer Science from Universidade Federal Rural de Pernambuco, Brazil, in 2009, and Universidade Federal de Pernambuco, Brazil, in 2011, respectively. He is currently working towards the Ph.D degree in Computer Science at the Universidade Federal de Pernambuco, Brazil. His current research interests include quantum computation, weightless neural networks and hybrid neural systems.

Wilson R. de Oliveira received the B.Sc (1982), M.Sc (1985) and Ph.D (2004) in Computer Science at Universidade Federal de Pernambuco (Brazil) specialising in Computing Theory of Artificial Neural Networks and Automata Theory. Sabbatical leave at the School of Computer Science, University of Birmingham, UK (2008), working on Matroid Theory and Discrete Differential Geometry. Has been working recently on Quantum Weightless Neural Networks, Combinatorial Hopf Algebras, Matroid Representations and Non-linearity and Chaos in Natural Temporal Series. Has published over a 50 articles in scientific journals and conferences. Joined the "Departmento de Estatística e Informática" at the "Universidade Federal Rural de Pernambuco" in 2000 where is now Associate Professor.



Teresa B. 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 a 200 articles in scientific journals and conferences, three books in Neural Networks and organised two of the Brazilian Symposium on Neural Networks. She is one of the editors-in-Chief of the International Journal of Computation Intelligence and Applications. Her research interests include weightless Neural Networks, hybrid neural systems and applications of Neural Networks.

Part III

Weightless neural network parameters and architecture selection in a quantum computer

Weightless neural network parameters and architecture selection in a quantum computer

Adenilton J. da Silva[a,b], Wilson R. de Oliveira[a], Teresa B. Ludermir[b]

[a]*Departamento de Estatística e Informática*
Universidade Federal Rural de Pernambuco
[b]*Centro de Informática*
Universidade Federal de Pernambuco

Abstract

Training artificial neural networks requires a tedious empirical evaluation to determine a suitable neural network architecture. To avoid this empirical process several techniques have been proposed to automatise the architecture selection process. In this paper, we propose a method to perform parameter and architecture selection for a quantum weightless neural network (qWNN). The architecture selection is performed through the learning procedure of a qWNN with a learning algorithm that uses the principle of quantum superposition and a non-linear quantum operator. The main advantage of the proposed method is that it performs a global search in the space of qWNN architecture and parameters rather than a local search.

Keywords:
Quantum neural networks, quantum weightless neural networks, quantum learning, architecture selection

1. Introduction

The exponential reduction of computers components know as Moore's law took computation from the classical physical domain to the quantum physics. The idea of quantum computation was initially proposed in [1], where Feynman states that quantum computers can simulate quantum physical systems exponentially faster than classical computers. Some quantum algorithms also overcome the best knew classical algorithms; the most famous examples being the Shor's factoring algorithm [2] that is exponentially faster than the best know classical algorithm and the Grover's search algorithm [3] with quadratic gain in relation to the best classical algorithm for unordered search. It is true that quantum computers are not yet a reality, but there has been an explosion of investment in quantum computing, the result of which are numerous proposals for quantum computers and the general belief which soon they will be realised. The use of an adiabatic quantum system with 84 quantum bits is reported in [4] and in [5] is reported the creation of a quantum system with 14 quantum bits. Empirical evaluations of ideas presented in this work for real problems requires a quantum computer with capacity to manipulate hundreds of qubits which is impossible with current technology.

One of the main characteristics of quantum computation is the quantum parallelism that for some problems allows quantum algorithms to have a speedup in relation to the classical algorithms. With quantum parallelism is possible to calculate all possible 2^n values of a $n-$ary Boolean function in a single query. However, we cannot visualise these outputs directly. A quantum measurement is necessary and it returns probabilistically only a more restrict value. The quantum algorithm design problem is then to perform quantum operations to increase the probability of the desired output.

Designing quantum algorithms is not an intuitive task. Attempts to bring quantum computing power to a greater range of problems is the development of quantum machine-learning algorithms as decision trees [6], evolutionary algorithms [7] and artificial neural networks [8, 9, 10, 11, 12, 13, 14, 15, 16]. In this paper, we are concerned in the field of quantum weightless neural networks.

Weightless neural networks (WNN) are not the most used model of artificial neural networks. WNN have been proposed by Aleksander [17] as engineering tools to perform pattern classification. Applications of WNN are described in several works [18, 19, 20, 21] and quantum versions of WNN have been proposed in [10, 15, 13].

The idea of quantum neural computation have been proposed in the nineties [22], since then several models of quantum neural networks have been proposed. For instance, quantum weightless neural networks [13], neural networks with quantum architecture [8] and a simple quantum neural network [11]. In all these works [13, 8, 11] a quantum neural network configuration is represented by a string of qubits and quantum learning algorithms are proposed within a common framework. The main idea of the learning algorithms in [13, 8, 11] is to present input data to all possible neural networks for a given architecture in superposition and perform a quantum search in the resulting superposition. The objective of this paper is to generalise this idea to allow architecture selection trough the training of a quantum weightless neural network. To achieve this objective we use a quantum weightless neural network that stores representations of weightless neural networks with different architectures in its memory positions and we define a quantum learning algorithm using the non-linear operator proposed in [23] and the measurement and feedback strategy [24].

Selection of a neural network architecture is an important

task in neural networks applications. Normally this task requires a lot of empirical evaluation performed by an expert. To avoid the tedious empirical evaluation process and help inexperienced users some algorithms have been proposed to perform automatic selection of neural networks architecture. Techniques such as meta-learning [25] and evolutionary computation [26] have been used for architecture selection.

In this paper, we show how to use a quantum weightless neural network with a non-linear quantum-learning algorithm to find a quantum neural network architecture and parameters with a desired performance. The proposed algorithm uses quantum superposition principle and a non-linear quantum operator. The proposed algorithm performs a global search in architecture and parameters space and its computational time is polynomial in relation to the number of training patterns, architectures and quantum weightless network memory size.

The rest of the paper is organised as follows. Section 2 presents basics concepts on quantum computation such as quantum bits, operators, measure and parallelism. Section 3 presents the concept of weightless neural networks, quantum neural networks and quantum weightless neural networks. Section 4 describes a quantum learning algorithm for weightless neural networks and how to apply this learning algorithm to perform architecture selection. Finally, Section 5 is the conclusion.

2. Quantum computing

Deep knowledge of classical physics is not required for designing classical algorithms. In the same vein, the development of quantum algorithms does not require a deep knowledge of quantum physics and there are several books [27, 28, 29] that follow this approach by introducing only the strictly necessary knowledge of quantum physics for the understanding of quantum computing. In order to create a self-contained text a brief introduction to quantum computing is presented.

The state of a quantum computer with n quantum bits (or *qubits*) can be mathematically represented by a unit vector of an 2^n-dimensional complex vector space with inner product. For instance, one single qubit can be represented in the vector space $\mathbb{C}^2$ as described in Equation (1),

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

where $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$ and $|0\rangle$ and $|1\rangle$ are the vectors described in Equation (2)[1].

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{2}$$

One qubit in a n-dimensional quantum system is represented by the 2^n-dimensional vector space as described in Equation (3),

$$\sum_{i=0}^{2^n-1} \alpha_i\,|\psi_i\rangle \tag{3}$$

where the sum of the squared modulus of the amplitude $\sum_i |\alpha_i|^2$ is equal to one and the set $\{|\psi_0\rangle, |\psi_1\rangle, \cdots, |\psi_{2^n-1}\rangle\}$ is an orthonormal basis of $\mathbb{C}^{2^n}$.

A *Quantum operator* in a quantum system with n qubits is an unitary operator[2] in the vector space $\mathbb{C}^{2^n}$. Let U be an unitary operator over $\mathbb{C}^{2^n}$ and $\left|\psi_{t_1}\right\rangle$ the state of the quantum system. After applying the quantum operator U the system will be in the state $\left|\psi_{t_2}\right\rangle$ described in Equation (4).

$$\left|\psi_{t_2}\right\rangle = U\left|\psi_{t_1}\right\rangle \tag{4}$$

In the computational basis, the matrix representation of the quantum operators the *not operator X* and the *Hadamard operator H* over one qubit are described in Equation (5).

$$X = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \text{ and } H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} \tag{5}$$

X acts on the computational basis vectors as a not operator ($X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$) and H applied to a state in the computational basis creates a "uniform" superposition (or linear combination) of the two basis:

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

both represent a state which is $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$, and can be thought of a state which is both $|0\rangle$ and $|1\rangle$. That is why one says that a qubit is able to "store" two classical bits simultaneously. This scale up exponentially with the number of qubits n, $H|0\rangle \otimes \cdots \otimes H|0\rangle = H^{\otimes n}|0\cdots0\rangle$, with $0\cdots0$ being a sequence of n 0's, is the superposition of all 2^n possibles n-qubits. Equation (7) shows the result for $n = 2$, where $H^{\otimes 2} = H \otimes H$:

$$\begin{aligned} H^{\otimes 2}|0\rangle|0\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) =\\ \frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle)\end{aligned} \tag{7}$$

Quantum parallelism is one of the main properties of quantum computation and it is used in the majority of quantum algorithms. Let U_f be a quantum operator with action described in Equation (8),

$$U_f|x, c\rangle = |x, c \oplus f(x)\rangle \tag{8}$$

where $f : B^m \to B^n$ is a Boolean function. Applying this operator in a state in superposition $\sum_i |x_i, 0\rangle$ the value of x_i will be calculated for all i in a single quantum operation, as described in Equation (9).

$$U_f\left(\sum_i |x_i, 0\rangle\right) = \sum_i U_f|x_i, 0\rangle = \sum_i |x_i, f(x_i)\rangle \tag{9}$$

[1] We could have used any other orthonormal basis but in quantum computing the canonical basis also called *computational basis* is the most employed.

[2] An operator (or matrix, for the finite dimensional case once fixed a basis) A is *unitary* if $AA^\dagger = A^\dagger A = I$ where $A^\dagger$ is the complex conjugate of the transpose of A

Despite the possibility of obtaining all possible outputs of a Boolean function in a single query, quantum parallelism cannot be used directly.

Results in quantum computation are obtained via *measurement* which returns only a limited information about the system. For instance, if a measurement is performed in a quantum state $|\psi\rangle = \alpha_i |\psi_i\rangle$ the result will be $|\psi_i\rangle$ with probability $|\alpha_i|^2$. After measurement state $|\psi\rangle$ collapses to the output obtained and new measurements will result in the same output.

With the definition given above, also adopted by the mainstream quantum literature as [27], quantum operators are linear operators. In this paper, we suppose the viability of a nonlinear quantum operator Q proposed in [23] whose action is described in Equation (10) if at least one $|c_i\rangle$ is equal to $|1\rangle$ otherwise its action is described in Equation (11).

$$Q\left(\sum_i |\psi_i\rangle |c_i\rangle\right) = \left(\sum_i |\psi_i\rangle\right)|1\rangle \tag{10}$$

$$Q\left(\sum_i |\psi_i\rangle |c_i\rangle\right) = \left(\sum_i |\psi_i\rangle\right)|0\rangle \tag{11}$$

The speedup obtained by the application of non-linear operators have been associated with unphysical effects, however in [30, 31] it is presented a version of this non linear quantum operator free of unphysical influences.

3. Classical and quantum weightless neural networks

This work deals with quantum weightless neural networks. Weightless Neural Networks (WNN) are neural networks without weights associated in their connections where the information is stored in a look up table. The first model of WNN named RAM have been proposed in [17], since then several neural networks models have been proposed, for instance the Probabilistic Logic Neuron (PLN), the Multi-valued Probabilistic Logic Neuron (MPLN), Goal Seeking Neuron (GSN) and the quantum RAM neuron (qRAM).

A weightless neuron with n input values has a memory with 2^n addressable positions. The learning procedure of a WNN does not require differential calculus or any complex mathematical calculations. The learning procedure is performed by writing in the look up table. This learning strategy is faster than techniques based in gradient descendant methods and are suitable to implementation in conventional digital hardware.

Several models of weightless neural networks are described in [32]. In this paper we deal with the qRAM neural network. The qRAM neuron is based in the simplest weightless model, the RAM neuron. Besides its simplicity RAM neurons can be trained very rapidly. Some applications of RAM and RAM based neurons in real world problems are described e.g. in [33, 20, 19, 34]. For a recent review see [21]. In [33, 19] a WiSARD system is used to track moving objects or human beings, in [20] a WiSARD clustering version is proposed to perform credit analysis and in [34] a VG-RAM weightless neural network is used to perform multi-label text categorisation.

3.1. RAM Node

A RAM neuron with n inputs has a memory C with 2^n addressable positions. Each memory position of a RAM neuron stores a Boolean value and its address is a Boolean string in $\{0,1\}^n$ also called Boolean vector. When a RAM neuron receives a Boolean vector $x = x_1 \cdots x_n$ as input it will produce the output $C[x]$. Learning in the qRAM node is very simple and can be accomplished updating the bits in memory positions for each one of the patterns in the training set.

Architectures of weightless neural networks are weekly connected as a consequence of the limited number of neurons inputs. Two common architecture are pyramidal where the output of a neuron in one layer is connected with a single neuron in the next layer or with only one layer where the neural network output is the sum of each neuron outputs.

3.2. Quantum neural networks

The notion of quantum neural networks have been proposed on several occasions [8, 11, 12]. In [8, 11] quantum neural models are pure abstract mathematical devices and in [12] quantum neural networks is described as a physical device. In this paper we follow the first approach where a neural network is a mathematical model. It is also possible to classify quantum neural networks models as either quantum neural model [35, 36, 8, 11, 12, 13] or quantum inspired model [37, 38]. Quantum inspired models are classical models of computation that uses ideas from quantum computing. Implementation of the quantum weightless neural network mathematically described in this paper requires a real quantum computer. Recent reviews on quantum neural networks can be found in [39, 40].

Models of quantum weightless neural networks are proposed or analysed in [10, 15, 13, 41]. Quantum weightless neural networks models are first proposed in [10], in [42] a quantum version of the RAM neuron based in an associative quantum memory is presented and in [13] the qRAM neuron and a learning algorithm for quantum weightless neural networks are presented.

Learning algorithms for quantum neural networks are also proposed in [8, 11] where a superposition of neural networks with a fixed architecture is created and a quantum search is performed to recover the best neural network architecture. In this paper we propose a variation of this methodology to train quantum weightless neural networks. In our training strategy, weightless neural networks with different architectures are in a superposition. The neural network model used in this learning methodology is the qRAM neural network.

3.3. qRAM Node

The qRAM neuron is a quantum version of the RAM neuron. As in classical case, a n input qRAM neuron has a quantum memory with 2^n memory positions. The content of the qRAM memory cannot be directly stored because a measurement of the output could destroy the information stored in the qRAM memory. We store quantum bits in the computational basis named selectors and apply one quantum operator A to obtain the stored qubit.

The A operator used in the qRAM is the control X operator described in Equation 12. With this operator a quantum RAM neuron can be described as in Definition 1, where memory contents are stored in quantum register selectors.

$$A = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} \quad \begin{array}{l} \text{where} \\ A\,|00\rangle = |0\rangle\, I\,|0\rangle \\ A\,|10\rangle = |1\rangle\, X\,|0\rangle \end{array} \tag{12}$$

Definition 1. *A qRAM node with n inputs is represented by the operator* N *described in Equation* (13)*. The inputs, selectors and outputs of* N *are organised in three quantum registers* $|i\rangle$ *with n qubits,* $|s\rangle$ *with* 2^n *qubits and* $|o\rangle$ *with 1 qubit. The quantum state* $|i\rangle$ *describe qRAM input, and quantum state* $|s\rangle\,|o\rangle$ *describes qRAM state.*

$$\mathsf{N} = \sum_{i=0}^{2^n-1} |i\rangle_n \langle i|_n A_{s_i,o} \tag{13}$$

The qRAM neural network functions exactly as a RAM neural network when the selectors are in the computational basis. For instance, when the quantum register selectors of a qRAM neuron is in the state $|c_{00}c_{01}c_{10}c_{11}\rangle$ and the input $|xy\rangle$ is presented its output is $\left|c_{xy}\right\rangle$. The difference between the qRAM and RAM neurons can be observed when the selectors are initialised with a state in superposition. Suppose an initialisation of the quantum register selector with a state in the superposition $\frac{1}{\sqrt{2}}\left(|c_{00}c_{01}c_{10}c_{11}\rangle + \left|c'_{00}c'_{01}c'_{10}c'_{11}\right\rangle\right)$. When the neuron receives an input $|xy\rangle$, the output for each configuration in the superposition will be calculated and the quantum register output will be in the state $\frac{1}{\sqrt{2}}\left(\left|c_{xy}\right\rangle + \left|c'_{xy}\right\rangle\right)$, a sort of parallel execution of the network.

Classical and quantum weightless neurons require a memory (in the classical case) and a number of selectors (in the quantum case) exponential in relation to the number of inputs. To avoid exponential memory requirements, classical and quantum weightless neural networks use a feed-forward, low connected, pyramidal architecture. A pyramidal, feed-forward neural network with three two inputs qRAM Nodes is shown in Figure 1. A pyramidal qRAM network with n inputs and composed of neurons with two inputs will have $2^{log_2(n)} - 1 = n - 1$ neurons. Each two input neuron has a memory with 4 selectors than network memory will need of $4 \cdot (n-1)$ selectors (linear memory size instead of an exponential memory size).

Configuration of a qRAM neural network is realised by the neuron selectors. For instance the configuration of the qRAM neural network in Figure 1 is the state of quantum registers $|s_1, s_2, s_3\rangle$. For instance, a qRAM network with architecture displayed in Figure 1 with configuration $|s_1\rangle = |0110\rangle$, $|s_2\rangle = |0110\rangle$ and $|s_3\rangle = |0110\rangle$ can solve the 4 bit parity problem. Superposition of qRAM neural networks with a given architecture can be obtained with the initialisation of qRAM neural configuration with a state in superposition. In Section 5, we explore superposition of qRAM networks in the learning procedure to allow neural network architecture selection.

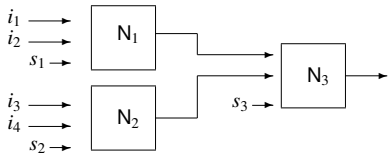


Figure 1: qRAM Neural Network of 2 layers

4. Non linear quantum learning

Nonlinear quantum operators have been used previously [8, 43]. In this section we show how to train a weightless neural network with a nonlinear quantum algorithm. The proposed algorithm is based in a strategy proposed in [24], where the learning procedure is performed by measurement and feedback. Figure 2 illustrates how the measurement and feedback strategy works. The input is presented to a controlled quantum operator named quantum processor, and the result of a measurement performed in the output registers is used to update qubits in the control quantum register. The procedure is repeated until the control qubits $|s\rangle$ are set to some desired value.

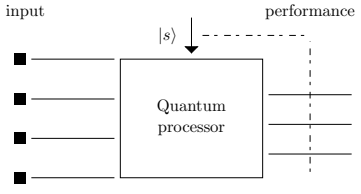


Figure 2: Measurement and feedback methodology

The quantum processor in our learning strategy will be a qRAM weightless neural network with a fixed architecture. This quantum weightless neural network can have any number of layers and neurons and must have a feed-forward architecture. Patterns selected from a training set will be presented to several neural networks in parallel. This step cannot be efficiently performed in a classical computer, but it can be performed in a quantum computer using quantum parallelism. Operation performed by the quantum processor is described in Figure 3 where each pattern x is presented to all qRAM network configurations represented in the quantum register $|s\rangle$ and the performance quantum register is updated to indicate if the neural network output is equal to the desired output $d(x)$. After the presentation of all patterns in the training set all pairs of neural network configuration and its respective performance will be in superposition.

Control qubits of the quantum processor in the measurement and feedback strategy are selectors of the qRAM neural network. In the kth iteration of the measurement and feedback methodology a non-linear quantum operator and a measure-

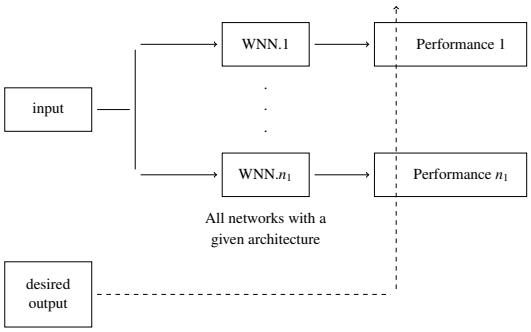


Figure 3: Action of quantum processor in Figure 2 when the selector quantum register of a qRAM weightless neural network with fixed architecture is a superposition of quantum states

ment are performed to determine the kth quantum bit of the selectors quantum register $|s\rangle$. After all iterations, the quantum register $|s\rangle$ will hold a qRAM configuration with performance greater than or equal to a given threshold θ for given training set (if exists).

Algorithm 1 presents the proposed learning strategy. It requires six quantum registers. Input quantum register $|i\rangle$ used to present patterns from the training set to the qRAM network. The free parameters or selectors quantum register $|s\rangle$ used to store qRAM neural network configuration. The output quantum register $|o\rangle$ used to store the qRAM neural network output, the desired output quantum register $|d\rangle$, performance quantum register $|perf\rangle$ used to store the performance of each classifier in the superposition. And the objective quantum register $|obj\rangle$ used to mark configurations with desired performance. A configuration of the weightless neuron during the execution of Algorithm 1 will be represented using the quantum state $|\psi\rangle$ described in Equation (14).

$$|\psi\rangle = |i\rangle\,|s\rangle\,|o\rangle\,|d\rangle\,|perf\rangle\,|obj\rangle \tag{14}$$

The for loop starting in line 1 will be repeated n_s times, where n_s is the number of quantum bits in quantum register $|s\rangle$. At the end of the kth iteration a non-linear quantum operator is performed to determine the kth bit l_k of the quantum register $|s\rangle$.

Steps 2, 4, 5, 6 initialise quantum registers input, output, performance and objective. Step 3 of Algorithm 1 initialises selector quantum register. After this step, the state of quantum registers $|s\rangle$ is described in Equation (15), where the value of first k qubits l_i were determined in ith iteration of the for loop and the last $n_s - k$ qubits are initialised with $H|0\rangle$ state.

$$|s\rangle = \left(\frac{1}{\sqrt{2}}\right)^{n_s-k+1} |l_1 \cdots l_{k-1}\rangle\,(|0\rangle + |1\rangle)^{\otimes(n_s-k+1)} \tag{15}$$

The for loop starting in line 7 performs the quantum processor operation. It calculates the performance of all configurations in the superposition for the given architecture simultaneously due to principle of quantum parallelism. Step 8 initialises

Algorithm 1: Learning algorithm

```
1  for k = 1 to n_s do
2      Set input quantum register |i⟩ to |0⟩
3      Set the n_s − k + 1 last qubits in quantum register |s⟩ to
       H|0⟩
4      Set output quantum register to |0⟩
5      Set performance quantum register to |0⟩
6      Set objective quantum register to |0⟩
7      for each pattern x ∈ training set do
8          Set quantum register |i⟩ to |x⟩ and quantum register
           |d⟩ to |d(x)⟩
9          Allow the qRAM network to produce it output in
           quantum register |o⟩
10         if |o⟩ = |d(x)⟩ then
11             add 1 into quantum register |perf⟩
12         end
13         Remove |x⟩ and |d(x)⟩ of quantum registers |i⟩ and
           |d⟩
14     end
15     for l = 0 to 1 do
16         Set quantum register objective to |1⟩ if kth quantum
           bit in neuron representation is equal to l and
           performance is greater than a given threshold θ.
17         Apply the non-linear quantum operator NQ to
           quantum register objective.
18         if |objective⟩ = |1⟩ then
19             Perform a measurement in all quantum register
20             Set kth bit of quantum register selectors to l
21         end
22     end
23 end
```

quantum register input with a pattern x from the data-set, and desired output quantum register with the desired output of x named $d(x)$. These initialisation steps can be performed by unitary operators controlled by a classical system using the classical representation of x and $d(x)$. Step 9 runs the qRAM neural network and its output quantum register is set to the calculated output $y(x, s)$ for pattern x with neural network configuration s. Steps 10 to 11 adds 1 to quantum register performance if $y(x, s)$ is equal to $d(x)$. After these steps, description of state $|\psi\rangle$ is presented in Equation (16), where state $|s\rangle$ is described in Equation (15) and $|perf(x, s)\rangle$ is the performance of the neural network with selectors s after reading the input x.

$$|\psi\rangle = |x\rangle\,|s\rangle\,|y(x,s)\rangle\,|d(x)\rangle\,|perf(x,s)\rangle\,|0\rangle \tag{16}$$

Step 13 removes $|x\rangle$ and $|d(x)\rangle$ of quantum registers $|i\rangle$ and $|d\rangle$ performing the inverse operation of Step 8. After the execution of the for loop starting in line 7 the performance of each classifier $|perf(s)\rangle$ will be in superposition with its representation $|s\rangle$.

The for loop starting in line 15 performs the measurement and feedback. An exhaustive non-linear quantum search is performed to determine the value of the kth bit in quantum state $|s\rangle$. Step 16 sets the quantum register $|obj\rangle$ to $|1\rangle$ if $perf(s) = \theta$

and $k = l$. This step can be performed by a unitary controlled operator U_g that flips objective quantum register if and only if $perf(x, s) \geq \theta$ and $k = l$. After Step 16 the state of quantum registers $|s\rangle$, $|perf\rangle$ and $|obj\rangle$ is described in Equation (17), where $\delta_{ms,l,perf(ms)}$ is equal to 1 if $perf(s) \geq \theta$ and the kth quantum bit in $|s\rangle$ is equal to l.

$$|s, perf, obj\rangle = \left|s, perf(s), \delta_{s,l,perf(s)}\right\rangle \quad (17)$$

All previous steps can be performed utilising only linear quantum operators. Step 17 applies the non-linear quantum operator proposed in [23] to the objective quantum register. The objective quantum register will be changed to the basis state $|1\rangle$ if there is at least one configuration in the superposition with objective equal to one. In this case, Steps 18 to 20 performs a measurement in state $|\psi\rangle$ and changes the kth quantum bit in quantum register $|s\rangle$ to l.

The computational cost of Algorithm 1 depends on the number of patterns in the training set n_t and on the number of qubits used in selector quantum register n_s. The for loop starting in line 1 will be repeated n_s times. Steps 2 to 6 have constant computational time. For loop in lines 7 to 13 will be repeated n_t times and each inner line has constant computational cost. For loop in lines 15 to 22 does not depend on n_t and n_s and it has a constant computational cost. In this way the overall cost of the Algorithm 1 is $O(n_t \cdot n_s)$. Then Algorithm 1 has polynomial time in relation to the number of qubits used to represent the qRAM neural network selectors and the number of patterns in the training set.

A concrete example of Algorithm 1 execution is presented to illustrate its functionality. Without loss of generality we use a qRAM neural network composed by only one neuron with two inputs to learn the 2-bit XOR toy problem described in Equation (18). For this problem, quantum register input needs two qubits, quantum register selectors has 4 qubits, quantum register output needs 1 qubit, quantum register performance has 3 qubits and quantum register objective has 1 qubit.

$$T = \{(|00\rangle, |0\rangle), (|01\rangle, |1\rangle), (|10\rangle, |1\rangle), (|11\rangle, |0\rangle)\} \quad (18)$$

In Steps 2, 4, 5 and 6 bits in quantum registers input, output, performance and objective are initialised with the quantum state $|0\rangle$. The number of quantum bits in $|s\rangle$ quantum register is equal to 4 and in the first iteration $n_s - k + 1$ is also equal to 4, then all four qubits in quantum register $|s\rangle$ are initialised with the state $H|0\rangle$. After these initialisation steps, neural network configuration $|\psi\rangle$ is described in Equation (19).

$$\begin{aligned} |\psi\rangle = \frac{1}{4} |00\rangle (|0\rangle + |1\rangle)^{\otimes 4} |0\rangle |0\rangle |000\rangle |0\rangle = \\ \frac{1}{4} \sum_{j\in\{0,1\}^4} |00\rangle |j\rangle |0\rangle |0\rangle |000\rangle |0\rangle \end{aligned} \quad (19)$$

Suppose that in the first iteration of the for loop starting in line 1 x assumes value $|01\rangle$ and $d(x)$ is $|1\rangle$. Step 8 initialises pattern and desired output quantum register respectively to $|01\rangle$ and $|1\rangle$. This initialisation can be performed through CNOT operators applied to $|\psi\rangle$ resulting in state $|\psi_1\rangle$ described in Equation (20).

$$\frac{1}{4} \sum_{j\in\{0,1\}^4} |01\rangle |j\rangle |0\rangle |1\rangle |000\rangle |0\rangle \quad (20)$$

Step 9 runs the neural network and this output is calculated in quantum register $|o\rangle$. After this step we obtain the state $|\psi_2\rangle$ described in Equation (21), where j_1 is the qubit in memory position 01 and $\delta_{j_1,1} = 1$ if and only if $j_1 = 1$.

$$\frac{1}{4} \sum_{j\in\{0,1\}^4} |01\rangle |j\rangle \left|\delta_{j_1,1}\right\rangle |1\rangle |000\rangle |0\rangle \quad (21)$$

Step 10 to 11 check if desired output is equal to the calculated output, adding one to the performance quantum register if they are equal. The resulting state after Step 11 $|\psi_3\rangle$ is described in Equation (22). These steps can be performed using a unitary operator describing the qRAM neural network and a quantum operator that adds one to the quantum register performance with controls $|o\rangle$ and $|d\rangle$.

$$\begin{aligned} |\psi_3\rangle = \frac{1}{4} (&|01\rangle |0000\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ + &|01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle + |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \\ &+ |01\rangle |0001\rangle |0\rangle |1\rangle |000\rangle |0\rangle \end{aligned} \quad (22)$$

Step 13 removes the values of $|x\rangle$ and $|d(x)\rangle$ from quantum registers $|i\rangle$ and $|d\rangle$ allowing the initialisation of the next for loop iteration. After the for loop last execution only one configuration in superposition, with $|s\rangle = |0110\rangle$, has performance 100% and the selectors and performance quantum registers are described by quantum state in Equation (23), where $perf(j) < 4$ for all $j \neq 0110$.

$$|s, perf\rangle = \frac{1}{4} \left(|0110\rangle |4\rangle_3 + \sum_{j\in\{0,1\}^4, j\neq 0110} |j\rangle |perf(j)\rangle \right) \quad (23)$$

Setting θ to 100%, in the first iteration of the for loop ($l = 0$) in line 15, Step 16 changes objective register to $|1\rangle$ when the kth qubit of $|s\rangle$ is $|0\rangle$ and performance is θ. After Step 15 selectors, performance and objective quantum registers are described in Equation (11).

$$|s, perf, obj\rangle = \frac{1}{4} \left(|0110\rangle |4\rangle_3 |1\rangle + \sum_{j\in\{0,1\}^4, j\neq 0110} |j\rangle |perf(j)\rangle |0\rangle \right) \quad (24)$$

Step 17 applies the nonlinear quantum operator in objective quantum register and the state of selectors, performance and

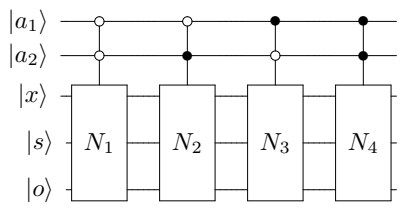


Figure 4: Quantum neuron representing a weightless neural networks with four different architectures

objective quantum registers are described in Equation (25). The nonlinear quantum operator sets the quantum register objective to $|1\rangle$.

$$|s, perf, obj\rangle = \frac{1}{4}\left(|0110\rangle|4\rangle_3 + \sum_{j\in\{0,1\}^4, j\neq 0110} |j\rangle|perf(j)\rangle\right)|1\rangle \tag{25}$$

Since the objective quantum register is in a base state we can check whether $|obj\rangle = |1\rangle$ with no information loss. In Steps 18 to 20 a measurement is performed in quantum register $|s\rangle$ and the first qubit of $|s\rangle$ is set to $|l_1\rangle = |0\rangle$. This qubit will not be changed in the next iterations.

At the end of the main for loop the selector quantum register $|s\rangle$ will be in the state $|0110\rangle$ and the desired configuration was found. Next section shows how to perform a search in the architecture space of a quantum weightless neural network.

5. Architecture learning

The operator A in a qRAM neural network is known as controlled not operator. In other models of quantum weightless neural networks this operator can assume different forms. For instance, in [10] the A operators of qPLN are represented in computational basis by the quantum operator A_{qPLN} described in Equation (26), where U is an arbitrary quantum operator

$$\begin{aligned} A_{qPLN} = |00\rangle\langle 00|\otimes \mathsf{I} + |01\rangle\langle 01|\otimes \mathsf{X} + \\ |10\rangle\langle 10|\otimes \mathsf{H} + |11\rangle\langle 11|\otimes \mathsf{U} \end{aligned} \tag{26}$$

and the A operators of a qMPLN with n qubits in each memory position are represented by the matrix described in Equation (27), where U_{p_k} is a rotation operator with angle p_k.

$$A_{qMPLN} = \sum_{k=0}^{n-1} |k\rangle\langle k|\otimes \mathsf{U}_{p_k} \tag{27}$$

These A operators are used to generate the values stored in a specific memory position. For instance in the qPLN, instead of storing the qubit $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, we store the qubits in the computational basis $|10\rangle$ and uses the operator A_{qPLN} to generate the content $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

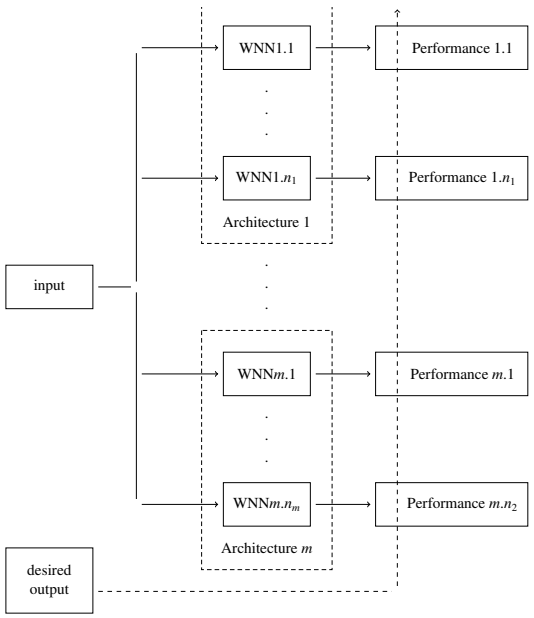


Figure 5: Action of quantum processor in Figure 2 when the selector and architecture selector quantum registers of a weightless neuron with some distinct architectures are in a superposition of quantum states

The main idea in this Section is to allow a weightless neural network to store the output of a weightless neural network with a given input x and selectors s. In this case, the quantum version of this weightless neural network will need a matrix A representing the weightless neural network to generate the output of the weightless neural network. Then the A operators are replaced by operators representing weightless neural networks and selectors are replaced by the neural network inputs and selectors. Figure 4 illustrates this weightless neuron with two inputs, where $|a_1a_2\rangle$ are architecture selectors, input pattern x and selectors are combined in one single quantum register and acts as the free parameters of the neuron, and quantum register output is shared by all weightless networks N_0, N_1, N_2, N_3.

With this quantum neuron the action of the quantum processor in Figure 2 can be described by Figure 5. Initialisation of the architecture selector quantum register with a quantum state in superposition will put different architectures, represented by the doted boxes, into superposition. And the initialisation of selectors quantum registers puts different configurations of each architecture into superposition. Problem of architecture selection is reduced to the problem of training the weightless neuron in Figure 4 where the input is represented by quantum register $|x\rangle$ and selectors are represented by quantum registers $|a, s\rangle$. In this way, Algorithm 1 can be used to learning parameters and architecture simultaneously.

Architecture selection computational time is directly related to computational time of the Algorithm 1. Due to the linearity

Patterns				Class
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Table 1: Simple artificial data set

of quantum operators, neurons can share selectors and under supposition that all architectures are pyramidal and low connected then network memory size (or the necessary number of selectors) will be polynomial in relation to the number of neural network inputs. The cost of architecture selection will be $O(n_a + n_s + n_t)$, where n_a is the number of architectures, n_s is the number of selectors in the most complex (with more selectors) architecture and n_t is the number of training patterns.

5.1. Architecture selection with SAL algorithm

Quantum computers are not yet a reality and we cannot evaluate SAL algorithm in real problems. In this Section we present a concrete example (with low dimensionality) of the SAL algorithm in architecture selection. Weuse the artificial dataset described in Table 1obtained in the following way. Two weightless neural network architectures were defined and an exhaustive search was performed to find a dataset in each one architecture can learn the dataset and the other architecture cannot learn the dataset using selectors in the computational basis.

The architectures used in the experiment are two layers, pyramidal qRAM weightless neural networks. The first architecture N_0 has two qRAM neurons each with two inputs in the first layer and one qRAM neuron with two inputs in the second layer. Figure 1 displays architecture N_0. The second architecture N_1 has two qRAM neurons in the first layer where the first neuron has three inputs and the second neuron has one input and the second layer has one qRAM neuron with two inputs.

The first architecture needs of 12 quantum bits for representing selector quantum register, 4 quantum bits for representing input of the first layer, 2 quantum bits to represent the second layer input, and 1 quantum bit to representing the neural network output. In this way, the first architecture representation needs of 19 quantum bits. The second architecture needs of 14 quantum bits for representing selector quantum register and the same number of quantum bits used by the first architecture to represent neurons inputs and network output than the second architecture representation requires 21 quantum bits.

These two qRAM neural networks are represented in a single circuit with six quantum registers. Neurons inputs quantum register $|i\rangle$ with 6 quantum bits, selectors quantum register $|s\rangle$ with 14 quantum bits, output quantum register $|o\rangle$ with one qubit and architecture selector quantum register $|a\rangle$ with 1 qubit. Performance quantum register $|perf\rangle$ with 5 quantum bits. Output quantum register with 1 quantum bit.

The qRAM neural network with architecture N_0 uses all qubits in quantum registers selectors, input and output. The qRAM neural network with architecture N_1 uses all qubits in inputs and output quantum register and uses only 12 qubits in selectors quantum register. The architecture quantum register $|a\rangle$ is used to select the architecture. If $|a\rangle$ is equal to 0 the architecture 1 is used. If $|a\rangle$ is equal to 1 the architecture 2 is used.

After the initialization steps of Algorithm 1, the state of quantum registers $|a\rangle|s\rangle|perf\rangle$ is described in Equation (28), where $|a\rangle$ and $|s\rangle$ are in a superposition with all possible values and the quantum bits in performance quantum register are initialized with $|0\rangle$.

$$|a\rangle|s\rangle|perf\rangle = (|0\rangle + |1\rangle) \sum_{k\in\{0,1\}^{14}} |k\rangle|00000\rangle \tag{28}$$

After the datased presetation to the neural network performed in Steps 7 to 14 of Algorithm 1, the state of quantum registers $|a\rangle|s\rangle|perf\rangle$ is described in Equation (29), where $perf(k, N_i)$ is the performance of qRAM neural network with architecture N_i and selectors $|k\rangle$.

$$\begin{aligned} |a\rangle|s\rangle|perf\rangle = \\ |0\rangle\left(\sum_{k\in\{0,1\}^{12}} |k\rangle \mathsf{H}^{\otimes 2}|00\rangle\right)|perf(k, N_0)\rangle \\ + |1\rangle \sum_{k\in\{0,1\}^{14}} |k\rangle|perf(k, N_1)\rangle \end{aligned} \tag{29}$$

N_0 architecture cannot learn the dataset with 100% of accuracy and N_1 can learn the dataset with 100% of accuracy when its selectors are in the set

$$\begin{aligned} T = \{&|01010111, 01, 1101\rangle, |01010111, 10, 1110\rangle, \\ &|10101000, 01, 0111\rangle, |10101000, 10, 1011\rangle\}. \end{aligned} \tag{30}$$

In the second iteration of for loop starting in line 15, the quantum register objective is set to $|1\rangle$ if and only if the performance is greather than a given threshold θ. Here we use θ equal to 16 (100% of accuracy), after this operation the state of quantum registers $|a\rangle|s\rangle|perf\rangle|obj\rangle$ is described in Equation (31).

$$
\begin{aligned}
|a\rangle |s\rangle |perf\rangle |obj\rangle = \\
|0\rangle \left(\sum_{k\in\{0,1\}^{12}} |k\rangle \, \mathsf{H}^{\otimes 2} \, |00\rangle \right) |perf(k, N_0)\rangle \, |0\rangle \\
+ |1\rangle \sum_{k\in\{0,1\}^{14}, k\notin T} |k\rangle \, |perf(k, N_1)\rangle \, |0\rangle \\
+ |1\rangle \sum_{k\in T} |k\rangle \, |perf(k, N_1)\rangle \, |1\rangle
\end{aligned}
\tag{31}
$$

Step 17 applies the nonlinear quantum operator and the resultant state of quantum registers $|a\rangle |s\rangle |perf\rangle |obj\rangle$ is described in Equation (32), where a measurement can be performed and the architecture register will be in state $|1\rangle$ and the architecture N_1 was chosen.

$$
|a\rangle |s\rangle |perf\rangle |obj\rangle = |1\rangle \sum_{k\in T} |k\rangle \, |perf(k, N_1)\rangle \, |1\rangle \tag{32}
$$

5.2. Discussion

We proposed a methodology to select quantum neural network parameters and architecture using a quantum weightless neural networks in polynomial time in relation to the number of training patterns, architectures and neural network free parameters. The proposed algorithm, named Superposition based Architecture Learning (SAL), performs a non-linear global search in the space of weightless neural networks parameters and for a given data set returns a classifier with a desired performance θ or returns that there is no classifier otherwise.

A classical polynomial time algorithm to perform neural network architecture selection is not known. Classical techniques used to perform architecture selection are heuristics that do not guarantee to find an exact solution. Some strategies used to find near optimal neural networks architectures or parameters are evolutionary algorithms [26] and meta-learning [44]. Running time of evolutionary algorithms used in architecture selection are displayed in [44] and even in benchmark problems the running time of these classical strategies can vary from 3 to 400 minutes.

In the application of the SAL algorithm to perform architecture selection, if there is a solution in the space search then the solution will be found in polynomial time. SAL algorithm puts all neural network configurations with some architectures in superposition, the performance is calculated and a nonlinear operator is used to recover the configuration and architecture with desired performance. SAL algorithm is the first algorithm to perform quantum weightless neural network architecture selection in polynomial time in relation to the number of patterns, architectures.

Superposition principle allows the evaluation of neural networks architectures in a way that is not possible in classical neural networks. In a classical neural network the architecture evaluation is biased by a choice of neural network parameters. In SAL algorithm all neural network parameters are initialized with all parameters in superposition allowing the evaluation of neural network architecture without the bias of a given set of parameters.

The gain in computational time of the proposed strategy is a result of the use of non-linear quantum operator proposed in [23]. Despite non-linear quantum computing has been used in several works, there still remains some controversy with some authors claiming that non linear quantum operators are not physically realisable [23] while other researchers claiming otherwise [30].

Even if non-linear quantum operators do not become a reality, the proposed learning algorithm furnishes a framework for the development of linear quantum algorithms to perform neural network architecture selection. The proposed idea is to define a quantum weightless neural network such that its memory positions store configurations of neural networks with different architectures.

6. Conclusion

For some problems there are quantum algorithms wich are assyntotically faster than the known classical algorithms [3, 2, 45]. In this paper, we defined a quantum Superposition based Architecture Learning algorithm for weightless neural networks that finds architecture and parameters with polynomial time in relation to the number of training patterns, architectures and the size of the selectors quantum register. The proposed algorithm used the quantum superposition principle and a nonlinear quantum operator.

A linear version of the proposed algorithm is challenging research topic which is the subject of on going work. This linear version should be a quantum probabilistic algorithm, because the problem of training a weightless neural networks is a NP-complete problem. One could use the quantum processor to create a superposition of weightless neural networks with different architectures and to perform classical learning steps in these neural networks in superposition before performing the measurement and feedback.

Quantum weightless neural networks proposed in [10] are generalisation of the classical models based in a classical RAM memory. Another possible future work is the analysis of quantum memories [46, 47] for the development of weightless neural networks models. These quantum memories has an exponential gain in memory capacity when compared with classical memories.

Acknowledgements

This work is supported by research grants from CNPq, CAPES and FACEPE (Brazilian research agencies).

References

[1] R. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (1982) 467.

[2] P. W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM Journal on Computing 26 (5) (1997) 1484–1509.

[3] L. K. Grover, Quantum Mechanics Helps in Searching for a Needle in a Haystack, Phys. Rev. Lett. 79 (2) (1997) 325–328.

[4] Z. Bian, F. Chudak, W. G. Macready, L. Clark, F. Gaitan, Experimental determination of ramsey numbers, Physical Review Letters 111. doi:10.1103/PhysRevLett.111.130505.

[5] T. Monz, P. Schindler, J. T. Barreiro, M. Chwalla, D. Nigg, W. A. Coish, M. Harlander, W. Hansel, M. Hennrich, R. Blatt, 14-qubit entanglement: Creation and coherence, Physical Review Letters 106. arXiv:1009.6126, doi:10.1103/PhysRevLett.106.130506.

[6] E. Farhi, S. Gutmann, Quantum computation and decision trees (1998). doi:10.1103/PhysRevA.58.915.

[7] A. Malossini, T. Calarco, Quantum genetic optimization, IEEE Transactions on Evolutionary Computation 12 (2008) 231–241. doi:10.1109/TEVC.2007.905006.

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

[9] M. V. Altaisky, Quantum neural network, arXiv:quant-ph/0107012 (2001).

[10] 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.

[11] B. Ricks, D. Ventura, Training a Quantum Neural Network, in: Advances in Neural Information Processing Systems, Cambridge, MA, 2004.

[12] E. C. Behrman, L. R. Nash, J. E. Steck, V. G. Chandrashekar, S. R. Skinner, Simulations of Quantum Neural Networks, Information Sciences 128 (3-4) (2000) 257–269.

[13] 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.

[14] A. Narayanan, T. Menneer, Quantum artificial neural networks architectures and components, Information Sciences 128 (3-4) (2000) 231–255.

[15] W. de Oliveira, Quantum RAM Based Neural Netoworks., in: ESANN, 2009, pp. 22–24.

[16] C. Y. Liu, C. Chen, C. T. Chang, L. M. Shih, Single-hidden-layer feed-forward quantum neural network based on Grover learning, Neural Networks 45 (2013) 144–150.

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

[18] M. Staffa, M. de Gregorio, M. Giordano, S. Rossi, Can you follow that guy?, in: European Symposium on Artificial Neural Networks, 2014, pp. 511–516.

[19] R. L. de Carvalho, D. Carvalho, P. M. V. Lima, F. Mora-Camino, F. M. G. França, Online tracking of multiple objects using WiSARD, in: European Symposium on Artificial Neural Networks, 2014, pp. 541–546.

[20] D. Cardoso, D. Carvalho, D. Alves, D. Souza, H. Carneiro, C. Pedreira, P. M. V. Lima, F. M. G. França, Credit analysis with a clustering RAM-based neural classifier, in: European Symposium on Artificial Neural Networks, 2014, pp. 517–522.

[21] M. D. Gregorio, F. M. G. França, P. M. V. Lima, W. R. de Oliveira, Advances on weightless neural systems, in: 22th European Symposium on Artificial Neural Networks, 2014, pp. 497–504.

[22] S. C. Kak, On Quantum Neural Computing, Information Sciences 83 (3) (1995) 143–160.

[23] 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.

[24] S. Gammelmark, K. Mø lmer, Quantum learning by measurement and feedback, New Journal of Physics 11 (3) (2009) 33017.

[25] A. Abraham, Meta learning evolutionary artificial neural networks, Neurocomputing 56 (2004) 1 – 38.

[26] L. M. Almeida, T. B. Ludermir, A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks, Neurocomputing 73 (7–9) (2010) 1438 – 1450.

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

[28] M. Hirvensalo, Quantum computing, Springer-Verlag, 2003.

[29] N. D. Mermin, Quantum computer science: an introduction, Cambridge University Press, 2007.

[30] M. Czachor, Remarks on search algorithms and nonlinearity, acta physica slovaca 48 (1998) 157–162.

[31] M. Czachor, Notes on nonlinear quantum algorithms, arXiv preprint quant-ph/9802051.

[32] T. B. Ludermir, A. de Carvalho, A. P. Braga, M. C. P. de Souto, Weightless neural models: a review of current and past works, Neural Computing Surveys 2 (1999) 41–61.

[33] M. Staffa, M. de Gregorio, M. Giordano, S. Rossi, Can you follow that guy?, in: European Symposium on Artificial Neural Networks, 2014, pp. 511–516.

[34] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese, C. Badue, Automated multi-label text categorization with VG-RAM weightless neural networks, Neurocomputing 72 (2009) 2209–2217. doi:10.1016/j.neucom.2008.06.028.

[35] M. Andrecut, M. K. Ali, a Quantum Perceptron, International Journal of Modern Physics B 16 (04) (2002) 639–645.

[36] M. Panella, G. Martinelli, Neurofuzzy networks with nonlinear quantum learning, IEEE Transactions on Fuzzy Systems 17 (3) (2009) 698–710.

[37] P. Li, H. Xiao, F. Shang, X. Tong, X. Li, M. Cao, A hybrid quantum-inspired neural networks with sequence inputs, Neurocomputing 117 (2013) 81–90. doi:10.1016/j.neucom.2013.01.029.

[38] N. Kouda, N. Matsui, H. Nishimura, F. Peper, Qubit neural network and its learning efficiency, Neural Comput. Appl. 14 (2) (2005) 114–121.

[39] M. Schuld, I. Sinayskiy, F. Petruccione, The quest for a quantum neural network, Quantum Information Processing 13 (11) (2014) 2567–2586.

[40] M. Altaisky, N. Kaputkina, V. Krylov, Quantum neural networks: Current status and prospects for development, Physics of Particles and Nuclei 45 (6) (2014) 1013–1032. doi:10.1134/S1063779614060033.
URL http://dx.doi.org/10.1134/S1063779614060033

[41] A. J. da Silva, T. B. Ludermir, W. R. de Oliveira, On the Universality of Quantum Logical Neural Networks, in: 2012 Brazilian Symposium on Neural Networks, Ieee, 2012, pp. 102–106. doi:10.1109/SBRN.2012.44.

[42] A. da Silva, W. de Oliveira, T. Ludermir, A weightless neural node based on a probabilistic quantum memory, in: Neural Networks (SBRN), 2010 Eleventh Brazilian Symposium on, 2010, pp. 259–264. doi:10.1109/SBRN.2010.52.

[43] 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.

[44] P. B. Miranda, R. B. Prudêncio, A. P. de Carvalho, C. Soares, A hybrid meta-learning architecture for multi-objective optimization of {SVM} parameters, Neurocomputing 143 (0) (2014) 27 – 43.

[45] C. A. Trugenberger, Quantum Pattern Recognition, Quantum Information Processing 1 (6) (2002) 471–493.

[46] M. V. Altaisky, N. E. Kaputkina, Quantum hierarchic models for information processing, International Journal of Quantum Information 10 (02) (2012) 1250026.

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

Part IV

Quantum perceptron over a field

Quantum perceptron over a field

Adenilton José da Silva[1] and Teresa Bernarda Ludermir
Centro de Informática, Universidade Federal de Pernambuco
Recife, Pernambuco, Brasil
E-mail: ajs@deinfo.ufrpe.br, tbl@cin.ufpe.br

Wilson Rosa de Oliveira
[1]*Departamento de Estatística e Informática*
Universidade Federal Rural de Pernambuco
Recife, Pernambuco, Brasil
E-mail: wilson.rosa@gmail.com

In this work we propose a quantum neural network named quantum perceptron over a field (QPF). QPF is a direct generalization of a classical perceptron and solve some drawbacks found in previous models of quantum perceptrons. We also present a learning algorithm named Superposition based Architecture Learning algorithm (SAL) that optimizes the neural network weights and architectures. SAL searches for the best architecture in a finite set of neural network architectures with linear time over the number of patterns in the training set. SAL is the first learning algorithm to determine neural network architectures in polynomial time. This speedup is obtained by the use of quantum parallelism and a non linear quantum operator.

1. Introduction

The size of computer components reduces each year and quantum effects have to be eventually considered in computation with future hardware. The theoretical possibility of quantum computing initiated with Benioff [1] and Feynman[2] and the formalization of the first quantum computing model was proposed by Deutsch in 1985[3]. The main advantage of quantum computing over classical computing is the use of a principle called superposition, which allied with linearity of the operators allows for a powerful form of parallelism to develop algorithms more efficients than the known classical ones. For instance, the Grover's search algorithm[4] and Shor's factoring algorithm[5] overcome any known classical algorithm.

Quantum computing has recently been used in the development of machine learning techniques as quantum decision trees[6], artificial neural networks[7–9], associative memory[10; 11], and inspired the development of novel evolutionary algorithms for continuous optimization problems[12; 13]. There is an increasing interest in quantum machine learning and in the quantum neural network area[14]. This paper proposes a quantum neural network named Quantum Perceptron over a Field (QPF) and investigates the use of quantum computing techniques to design a learning algorithm for neural networks.

Artificial neural networks are a universal model of computation[15] and have several applications in real life problems. For instance, in the solution of combinatorial optimization problems[16], pattern recognition[17], but have some problems as the lack of an algorithm to determine optimal architectures[18], memory capacity and high cost learning algorithms[19].

Notions of Quantum Neural Networks have been put forward since the nineties[20], but a precise definition of what is a quantum neural network that in-

tegrates neural computation and quantum computation is a non-trivial open problem[14]. To date, the proposed models in the literature are really just quantum inspired in the sense that despite using quantum representation of data and quantum operators, in a way or another some quantum principles are violated usually during training. Weights adjustments needs measurements (observation) and updates.

Research in quantum neural computing is unrelated, as stated in Ref. 14:

> "QNN research remains an exotic conglomeration of different ideas under the umbrella of quantum information".

and there is no consensus of what are the components of a quantum neural network. Several models of quantum neural networks have been proposed and they present different conceptual models. In some models a quantum neural network is described as a physical device[7]; as a model only inspired in quantum computing[21]; or as a mathematical model that explores quantum computing principles[22; 8; 9; 23]. We follow the last approach and assume that our quantum neural network model would be implemented in a quantum computer that follows the quantum principles as *e.g.* described in Ref. 24. We assume that our models is implemented in the quantum circuit model of Quantum Computing[24].

Some advantages of quantum neural models over the classical models are the exponential gain in memory capacity[25], quantum neurons can solve nonlinearly separable problems [22], and a nonlinear quantum learning algorithm with polynomial time over the number of patterns in the data set is presented in Ref. 8. However, these quantum neural models cannot be viewed as a direct generalization of a classical neural network and have some limitations presented in Section 4. Quantum computing simulation has exponential cost in relation to the number of qubits. Experiments with benchmarks and real problems are not possible because of the number of qubits necessary to simulate a quantum neural network.

The use of artificial neural networks to solve a problem requires considerable time for choosing parameters and neural network architecture[26]. The architecture design is extremely important in neural network applications because a neural network with a simple architecture may not be capable to perform the task. On the other hand a complex architecture can overfitting the training data[18]. The definition of an algorithm to determine (in a finite set of architectures) the best neural network architecture (minimal architecture for a given learning task that can learn the training dataset) efficiently is an open problem. The objective of this paper is to show that with the supposition of non-linear quantum computing[8; 27; 28] we can determine an architecture that can learn the training data in linear time with relation to the number of patterns in the training set. To achieve this objective, we propose a quantum neural network that respect the principles of quantum computation, neural computing and generalizes the classical perceptron. The proposed neuron works as a classical perceptron when the weights are in the computational basis, but as quantum perceptron when the weights are in superposition. We propose a neural network learning algorithm which uses a non-linear quantum operator[8; 28] to perform a global search in the space of weights and architecture of a neural network. The proposed learning algorithm is the first algorithm performing this kind of optimization in polynomial time and presents a framework to develop linear quantum learning algorithms to find near optimal neural network architectures.

The remainder of this paper is divided into 5 Sections. In Section 2 we present preliminary concepts necessary to understand this work. In Section 4 we present related works. Section 5 presents main results of this paper. We define the new model of a quantum neuron named quantum perceptron over a field that respect principles of quantum and neural computing. Also in Section 5 we propose a quantum learning algorithm for neural networks that determines a neural network architecture that can learn the train set with some desired accuracy. Section 6 presents a discussion. Finally, Section 7 is the conclusion.

2. Preliminary concepts

As quantum neural networks is a multidisciplinary subject involving concepts from Computer Science, Mathematics and Physics in this section some background concepts in the field of Quantum Computing is presented for the sake of self-containment.

3. Quantum computing

In this Section, we perform a simple presentation of quantum computing with the necessary concepts to understand the following sections. As in theoretical classical computing we are not interested in how to store or physically represent a quantum bit. Our approach is a mathematical. We deal with how we can abstractly compute with quantum bits or design abstract quantum circuits and models. We take as a guiding principle that when a universal quantum computer will be at our disposal, we could implement the proposed quantum neural models.

We define a quantum bit as unit complex bidimensional vector in the vector space $\mathbb{C}^2$. In the quantum computing literature a vector is represented by the Dirac's notation $|\cdot\rangle$. The computational basis is the set $\{|0\rangle, |1\rangle\}$, where the vectors $|0\rangle$ and $|1\rangle$ can be represented as in Equation (1):

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{1}$$

A quantum bit $|\psi\rangle$, *qubit*, is a vector in $\mathbb{C}^2$ that has a unit length as described in Equation (2), where $|\alpha|^2 + |\beta|^2 = 1$, $\alpha, \beta \in \mathbb{C}$.

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

While in classical computing there are only two possible bits, 0 or 1, in quantum computing there are an infinitely many quantum bits, for a quantum bit can be a linear combination (or superposition) of $|0\rangle$ and $|1\rangle$.

Tensor product $\otimes$ is used to define multi-qubit systems. On the vectors, if $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|\phi\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, then the tensor product $|\psi\rangle \otimes |\phi\rangle$ is equal to the vector $|\psi\phi\rangle = |\psi\rangle \otimes |\phi\rangle = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle) = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$. Quantum states representing n qubits are in a 2^n dimensional complex vector space. On the spaces, let $\mathbb{X} \subset \mathbb{V}$ and $\mathbb{X}' \subset \mathbb{V}'$ be basis of respectively vector spaces $\mathbb{V}, \subset \mathbb{V}'$. The tensor product $\mathbb{V} \otimes \mathbb{V}'$ is the set vector space obtained from the basis $\{|b\rangle \otimes |b'\rangle; |b\rangle \in \mathbb{X} \text{ and } |b'\rangle \in \mathbb{X}'\}$. The symbol $\mathbb{V}^{\otimes n}$ represents a tensor product $\mathbb{V} \otimes \cdots \otimes \mathbb{V}$ with n factors.

Quantum operators over n qubits are represented by $2^n \times 2^n$ unitary matrices. An $n \times n$ matrix is unitary if $\mathsf{U}\mathsf{U}^\dagger = \mathsf{U}^\dagger\mathsf{U} = \mathsf{I}$, where $\mathsf{U}^\dagger$ is the transpose conjugate of U. For instance, identity I, the flip X, and the Hadamard H operators are important quantum operators over one qubit and they are described in Equation (3).

$$\mathsf{I} = \begin{bmatrix}1&0\\0&1\end{bmatrix} \mathsf{X} = \begin{bmatrix}0&1\\1&0\end{bmatrix} \mathsf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix} \tag{3}$$

The operator described in Equation (4) is the *controlled not* operator CNot, that flips the second qubit if the first (the controlled qubit) is the state $|1\rangle$.

$$\mathsf{CNot} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} \tag{4}$$

Parallelism is one of the most important properties of quantum computation used in this paper. If $f : \mathbb{B}^m \to \mathbb{B}^n$ is a Boolean function, $\mathbb{B} = \{0, 1\}$, one can define a quantum operator

$$\mathsf{U}_f : \left(\mathbb{C}^2\right)^{\otimes n+m} \to \left(\mathbb{C}^2\right)^{\otimes n+m}, \tag{5}$$

as

$$\mathsf{U}_f|x, y\rangle = |x, y \oplus f(x)\rangle,$$

where $\oplus$ is the bitwise xor, such that the value of $f(x)$ can be recovered as

$$\mathsf{U}_f|x, 0\rangle = |x, f(x)\rangle.$$

The operator U_f is sometimes called the *quantum oracle* for f. Parallelism occurs when one applies the operator U_f to a state in superposition as e.g. described in Equation (6).

$$\mathsf{U}_f\left(\sum_{i=0}^{n} |x_i, 0\rangle\right) = \sum_{i=0}^{n} |x_i, f(x_i)\rangle \tag{6}$$

The meaning of Equation (6) is that if one applies operator U_f to a state in superposition, by linearity, the value of $f(x_i)$, will be calculated for each i simultaneously with only one single application of the quantum operator U_f.

With quantum parallelism one can imagine that if a problem can be described with a Boolean function f then it can be solved instantaneously. The problem is that one cannot direct access the values in a quantum superposition. In quantum computation a measurement returns a more limited value. The measurement of a quantum state $|\psi\rangle = \sum_{i=1}^{n} \alpha_i|x_i\rangle$ in superposition will return x_i with probability $|\alpha_i|^2$. With this result the state $|\psi\rangle$ collapses to state $|x_i\rangle$, i.e., after the measurement $|\psi\rangle = |x_i\rangle$.

Quantum neural computation research started in the nineties[20], and the models (such as *e.g.* in Refs. 7, 8, 22, 29, 9, 30) are yet unrelated. We identify three types of quantum neural networks: 1) models described mathematically to work on a quantum computer (as in Refs. 29, 30, 22, 8, 31, 9, 32, 33); 2) models described by a quantum physical device (as in Refs. 7, 34); and 3) models only based (inspired) on quantum computation, but that works in a classical computer (as in Refs 21, 35, 36). In the following subsections we describe some models of quantum neural networks.

4.1. *qANN*

Altaisky[29] proposed a quantum perceptron (qANN). The qANN N is described as in Equation (7),

$$|y\rangle = \hat{F}\sum_{j=1}^{n}\hat{w}_j|x_j\rangle \tag{7}$$

where $\hat{F}$ is a quantum operator over 1 qubit representing the activation function, $\hat{w}_j$ is a quantum operator over a single qubit representing the j-th weight of the neuron and $|x_j\rangle$ is one qubit representing the input associated with $\hat{w}_j$.

The qANN is one of the first models of quantum neural networks. It suggests a way to implement the activation function that is applied (and detailed) for instance in Ref. 8.

It was not described in Ref. 29 how one can implement Equation (7). An algorithm to put patterns in superposition is necessary. For instance, the storage algorithm of a quantum associative memory[25] can be used to create the output of the qANN. But this kind of algorithm works only with orthonormal states, as shown in Proposition 1.

Proposition 1. *Let $|\psi\rangle$ and $|\theta\rangle$ be two qubits with probability amplitudes in $\mathbb{R}$, if $\frac{1}{\sqrt{2}}(|\psi\rangle + |\theta\rangle)$ is a unit vector then $|\psi\rangle$ and $|\theta\rangle$ are orthogonal vectors.*

Proof. Let $|\psi\rangle$ and $|\theta\rangle$ be qubits and suppose that $\frac{1}{\sqrt{2}}(|\psi\rangle + |\theta\rangle)$ is a unit vector. Under these conditions

$$\frac{1}{2}(|\psi\rangle + |\theta\rangle, |\psi\rangle + |\theta\rangle) = 1 \Rightarrow$$
$$\frac{1}{2}(\langle\psi|\psi\rangle + \langle\psi|\theta\rangle + \langle\theta|\psi\rangle + \langle\theta|\theta\rangle) = 1$$
$$\frac{1}{2}(2 + 2\langle\psi|\theta\rangle) = 1 \tag{8}$$

and $|\psi\rangle$ and $|\theta\rangle$ must be orthogonal vectors. □

A learning rule for the qANN has been proposed and it is shown that the learning rule drives the perceptron to the desired state $|d\rangle$[29] in the particular case described in Equation (9) where is supposed that $F = I$. But this learning rule does not preserve unitary operators[9].

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

In Ref. 32 a quantum perceptron named Autonomous Quantum Perceptron Neural Network (AQPNN) is proposed. This model has a learning algorithm that can learn a problem in a small number of iterations when compared with qANN and this weights are represented in a quantum operator as the qANN weights.

4.2. *qMPN*

Proposition 1 shows that with the supposition of unitary evolution, qANN with more than one input is not a well defined quantum operator. Zhou and Ding proposed a kind of quantum perceptron[22] which they called as quantum M-P neural network (qMPN). The weights of qMPN are stored in a single squared matrix W that represents a quantum operator. We can see the qMPN as a generalized single weight qANN, where the weight is a quantum operator over any number of qubits. The qMPN is described in Equation (10), where $|x\rangle$ is an input with n qubits, W is a quantum operator over n qubits and $|y\rangle$ is the output.

$$|y\rangle = W|x\rangle \tag{10}$$

They also proposed a quantum learning algorithm for the new model. The learning algorithm for qMPN is described in Algorithm 1. The weights update rule of Algorithm 1 is described in Equation (11),

$$w_{ij}^{t+1} = w_{ij}^{t} + \eta(|d\rangle_i - |y\rangle_i)|x\rangle_j \tag{11}$$

where w_{ij} are the entries of the matrix W and η is a learning rate.

Algorithm 1 Learning algorithm qMPN

1: Let $W(0)$ be a weight matrix
2: Given a set of quantum examples in the form $(|x\rangle, |d\rangle)$, where $|x\rangle$ is an input and $|d\rangle$ is the desired output
3: Calculate $|y\rangle = W(t)|x\rangle$, where t is the iteration number
4: Update the weights following the learning rule described in Equation (11).
5: Repeat steps 3 and 4 until a stop criterion is met.

The qMPN model has several limitations in respect to the principles of quantum computing. qMPN is equivalent to a single layer neural network and its learning algorithm leads to non unitary neurons as shown in Ref. 37.

4.3. *Neural network with quantum architecture*

In the last subsections the neural network weights are represented by quantum operators and the inputs are represented by qubits. In the classical case, inputs and free parameters are real numbers. So one can consider to use qubits to represent inputs and weights. This idea was used, for instance, in Refs. 30 and 8. In Ref. 30 a detailed description of the quantum neural network is not presented.

In Ref. 8 is proposed a Neural Network with Quantum Architecture (NNQA) based on a complex valued neural network named *qubit neural network*[21] which is not a quantum neural network being just inspired in quantum computing. Unlike previous models[29; 22], NNQA uses fixed quantum operators and the neural network configuration is represented by a string of qubits. This approach is very similar to the classical case, where the neural network configuration for a given architecture is a string of bits.

Non-linear activation functions are included in NNQA in the following way. Firstly is performed a discretization of the input and output space, the scalars are represented by Boolean values. In doing so a neuron is just a Boolean function f and the quantum oracle operator for f, U_f, is used to implement the function f acting on the computational basis

In the NNQA all the data are quantized with N_q bits. A synapses of the NNQA is a Boolean function $f_0 : B^{2N_q} \to B^{N_q}$ A synapses of the NNQA is a Boolean function (12).

$$z = arctan\left(\frac{\sin(y)+\sin(\theta)}{\cos(y)+\cos(\theta)}\right) \tag{12}$$

The values y and θ are angles in the range $[-\pi/2, \pi/2]$ representing the argument of a complex number, which are quantized as described in Equation (13). The representation of the angle β is the binary representation of the integer k.

$$\beta = \pi\left(-0.5 + \frac{k}{2^{Nq}-1}\right), k = 0, 1, \cdots, 2^{Nq}-1 \tag{13}$$

Proposition 2. *The set* $F = \{\beta_k | \beta_k = \pi \cdot (-0.5 + k/(2^{Nq}-1)), k = 0, \cdots, 2^{Nq}-1\}$ *with canonical addition and multiplication is not a field.*

Proof. We will only show that the additive neutral element is not in the set. Suppose that $0 \in F$. So,

$$\pi \cdot \left(-0.5 + \frac{k}{2^{Nq}-1}\right) = 0 \Rightarrow -0.5 + \frac{k}{2^{Nq}-1} = 0$$

$$\Rightarrow \frac{k}{2^{Nq}-1} = 0.5 \Rightarrow k = (2^{Nq}-1)\cdot\left(-\frac{1}{2}\right)$$

Nq is a positive integer, and $2^{Nq}-1$ is an odd positive integer, then $(2^{Nq}-1)\cdot\left(-\frac{1}{2}\right) \notin \mathbb{Z}$ which contradicts the assumption that $k \in \mathbb{Z}$ and so F is not a field since $0 \notin F$. □

From Proposition 2 we conclude that the we cannot directly lift the operators and algorithms from classical neural networks to NNQA. In a weighted neural network inputs and parameters are rational, real or complex numbers and the set of possible weights of NNQA under operations defined in NNQA neuron is not a field.

5. Quantum neuron

5.1. *Towards a quantum perceptron over a field*

In Definition 3 and Definition 4 we have, respectively, an artificial neuron and a classical neural network as in Ref. 38. Weights and inputs in classical artificial neural networks normally are in the set of real (or complex) numbers.

Definition 3. An *artificial neuron* is described by the Equation (14), where $x_1, x_2, \cdots, x_k$ are input signals and $w_1, w_2, \cdots, w_k$ are weights of the synaptic

links, $f(\cdot)$ is a nonlinear activation function, and y is the output signal of the neuron.

$$y = f\left(\sum_{j=0}^{m} w_j x_j\right) \tag{14}$$

In both qANN and qMPN artificial neurons, weights and inputs are in different sets (respectively in quantum operators and qubits) while weights and inputs in a classical perceptron are elements of the same field. The NNQA model defines an artificial neuron where weights and inputs are strings of qubits. The neuron is based on a complex valued network and does not exactly follow Definition 3. The main problem in NNQA is that the inputs and weights do not form a field with sum and multiplication values as we show in Proposition 2. There is no guarantee that the set of discretized parameters is closed under the operations between qubits.

Other models of neural networks where inputs and parameters are qubits were presented in Ref. 39, 40, 9. These models are a generalization of weightless neural network models, whose definitions are not similar to Definition 3.

Definition 4. A *neural network* is a directed graph consisting of nodes with interconnecting synaptic and activation links, and is characterized by four properties[38]:

(1) Each neuron is represented by a set of linear synaptic links, an externally applied bias, and a possibility non-linear activation link. The bias is represented by a synaptic link connected to an input fixed at +1.
(2) The synaptic links of a neuron weight their respective input signals.
(3) The weighted sum of the input signals defines the induced local field of the neuron in question.
(4) The activation link squashes the induced local field of the neuron to produce an output.

The architecture of NNQA can be viewed as a directed graph consisting of nodes with interconnecting synaptic and activation links as stated in Definition 4. The NNQA does not follow all properties of a neural network (mainly because it is based in the qubit NN), but it is one of the first quantum neural networks with weights and a well defined architecture. The main characteristics of the NNQA are that inputs and weights are represented by a string of qubits, and network follows a unitary evolution. Based on these characteristics we will propose the quantum perceptron over a field.

5.2. *Neuron operations*

We propose a quantum perceptron with the following properties: it can be trained with a quantum or classical algorithm, we can put all neural networks for a given architecture in superposition, and if the weights are in the computational basis the quantum perceptron acts like the classical perceptron. One of the difficulties to define a quantum perceptron is that the set of n-dimensional qubits, sum and (tensor) product operators do not form a field (as shown in Proposition 5). Therefore, the first step in the definition of the quantum perceptron is to define an appropriate sum and multiplication of qubits.

Proposition 5. *The set of qubits under sum + of qubits and tensor product $\otimes$ is not a field.*

Proof. The null vector has norm 0 and is not a valid qubit. Under + operator the null vector is unique, so there is not a null vector in the set of qubits. Then we cannot use + operator to define a field in the set of qubits.

Tensor product between two qubits, results in a compound system with two qubits. So the space of qubits is not closed under the $\otimes$ operator and we cannot use $\otimes$ operator to define a field in the set of qubits. □

We will define unitary operators to perform sum $\oplus$ and product $\odot$ of qubits based in the field operations. Then we will use these operators to define a quantum neuron. Let $(F, \oplus, \odot)$ be a finite field. We can associate the values $a \in F$ to vectors (or qubits) $|a\rangle$ in a basis of a vector space V. If F has n elements the vector space will have dimension n.

Product operation $\odot$ in F can be used to define a product between vectors in V. Let $|a\rangle$ and $|b\rangle$ be qubits associated with scalars a and b, we define $|a\rangle \odot |b\rangle = |a \odot b\rangle$ such that $|a \odot b\rangle$ is related with the scalar $a \odot b$. This product send basis elements to basis elements, so we can define a linear operator $\mathsf{P} : V^3 \to V^3$ as in Equation (15).

$$\mathsf{P}|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|c \oplus (a \odot b)\rangle \tag{15}$$

We show in Proposition 6 that the P operator is unitary, therefore P is a valid quantum operator.

Proposition 6. *P is a unitary operator.*

Proof.

Let $B = \{|a_1\rangle, |a_2\rangle, \cdots, |a_n\rangle\}$ be a computational basis of a vector space V, where we associate $|a_i\rangle$ with the element a_i of the finite field $F(\oplus, \odot)$. The set $B^3 = \{|a_i\rangle|a_j\rangle|a_k\rangle\}$ with $1 \leq i, j, k \leq n$ is a computational basis of vector space V^3. We will show that P sends elements of basis B^3 in distinct elements of basis B^3.

P sends vectors in base B^3 to vectors in base B^3. Let a, b, c in B. $\mathsf{P}|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|c \oplus (a \odot b)\rangle$. Operators $\oplus$ and $\odot$ are well defined, then $|c \oplus (a \odot b)\rangle \in B$ and $|a\rangle|b\rangle|c \oplus (a \odot b)\rangle \in B^3$.

P is injective. Let $a, b, c, a_1, b_1, c_1 \in F$ such that $\mathsf{P}|a\rangle|b\rangle|c\rangle = \mathsf{P}|a_1\rangle|b_1\rangle|c_1\rangle$. By the definition of operator P $a = a_1$ and $b = b_1$. When we apply the operator we get $|a\rangle|b\rangle|c \oplus (a \odot b)\rangle = |a\rangle|b\rangle|c_1 \oplus (a \odot b)\rangle$. Then $c_1 \oplus (a \odot b) = c \oplus (a \odot b)$. By the field properties we get $c_1 = c$.

An operator over a vector space is unitary if and only if the operator sends some orthonormal basis to some orthonormal basis[41]. As P is injective and sends vectors in base B^3 to vectors in base B^3, we conclude that P is an unitary operator. □

With a similar construction, we can use the sum operation $\oplus$ in F to define a unitary sum operator $\mathsf{S} : V^3 \to V^3$. Let $|a\rangle$ and $|b\rangle$ be qubits associated with scalars a and b, we define $|a\rangle \oplus |b\rangle = |a \oplus b\rangle$ such that $|a \oplus b\rangle$ is related with the scalar $a \oplus b$. We define the unitary quantum operator S in Equation (16).

$$\mathsf{S}|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|c \oplus (a \oplus b)\rangle \tag{16}$$

We denote product and sum of vectors over F by $|a\rangle \odot |b\rangle = |a \odot b\rangle$ and $|a\rangle \oplus |b\rangle = |a \oplus b\rangle$ to represent, respectively, $\mathsf{P}|a\rangle|b\rangle|0\rangle = |a\rangle|b\rangle|a \odot b\rangle$ and $\mathsf{S}|a\rangle|b\rangle|0\rangle = |a\rangle|b\rangle|a \oplus b\rangle$.

5.3. *Quantum perceptron over a field*

Using S and P operators we can define a quantum perceptron analogously as the classical one. Inputs $|x_i\rangle$, weights $|w_i\rangle$ and output $|y\rangle$ will be unit vectors (or qubits) in V representing scalars in a field F. Equation (17) describes the proposed quantum perceptron.

$$|y\rangle = \bigoplus_{i=0}^{n} |x_i\rangle \odot |w_i\rangle \tag{17}$$

If the field F is the set of rational numbers, then Definition 3 without activation function f correspond to Definition 17 when inputs, and weights are in the computational basis.

The definition in Equation (17) hides several ancillary qubits. The complete configuration of a quantum perceptron is given by the state $|\psi\rangle$ described in Equation (18),

$$\begin{aligned} |\psi\rangle = |x_1, \cdots, x_n, w_1, \cdots, w_n, \\ p_1, \cdots, p_n, s_1, \cdots, s_{n-1}, y\rangle \end{aligned} \tag{18}$$

where $|x\rangle = |x_1, \cdots, x_n\rangle$ is the input quantum register, $|w\rangle = |w_1, \cdots, w_n\rangle$ is the weight quantum register, $|p\rangle = |p_1, \cdots, p_n\rangle$ is an ancillary quantum register used to store the products $|x_i \odot w_i\rangle$, $|s\rangle = |s_1, \cdots, s_{n-1}\rangle$ is an ancillary quantum register used to store sums, and $|y\rangle$ is the output quantum register. From Equation (18) one can see that to put several or all possible neural networks in superposition one can simply put the weight quantum register in superposition. Then a single quantum neuron can be in a superposition of several neurons simultaneously.

A quantum perceptron over a finite d-dimensional field and with n inputs needs $4 \cdot n \cdot d + d$ quantum bits to perform its computation. There are n quantum registers to store inputs x_i, n quantum registers to store weights w_i, n quantum registers p_i to store the products $w_i \odot x_i$, n quantum registers $|s_i\rangle$ to store sums and one quantum register to store the output $|y\rangle$.

We show now a neuron with 2 inputs to illustrate the workings of the quantum neuron. Suppose $F = \mathbb{Z}_2 = \{0, 1\}$. As the field has only two elements we need only two orthonormal quantum states to represent the scalars. We choose the canonical ones $0 \leftrightarrow |0\rangle$ and $1 \leftrightarrow |1\rangle$.

Now we define sum $\oplus$ and multiplication $\odot$ operators based on the sum and multiplication in $\mathbb{Z}_2$. The operators S and P are shown, respectively, in Equations (19) and (20).

$$S = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \tag{19}$$

$$P = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{bmatrix} \tag{20}$$

Using S and P operators we describe the quantum neuron N in Equation (21). The subscripts in operators indicate the qubits upon which they will be applied.

$$\mathsf{N} = \mathsf{S}_{p_1p_2y}\mathsf{P}_{x_2w_2,p_2}\mathsf{P}_{x_1w_1,p_1} \tag{21}$$

For our execution example, we define $|x_1x_2\rangle = |01\rangle$, $|w_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|w_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. The initial configuration of the quantum perceptron is $|\psi_0\rangle$ described in Equation (22). The initial configuration has all possible weights in the set $\{0,1\}^2$ and applying the QP will result the output for each weight simultaneously.

$$\begin{aligned} |x_1x_2\rangle|w_1w_2\rangle|p_1p_2\rangle|s\rangle|y\rangle = \\ \frac{1}{2}|01\rangle\left(|00\rangle + |01\rangle +\right)|00\rangle|0\rangle|0\rangle = \\ \frac{1}{2}\left(|01\rangle|00\rangle|00\rangle|0\rangle|0\rangle + |01\rangle|01\rangle|00\rangle|0\rangle|0\rangle + \right. \\ \left. |01\rangle|10\rangle|00\rangle|0\rangle|0\rangle + |01\rangle|11\rangle|00\rangle|0\rangle|0\rangle\right) \end{aligned} \tag{22}$$

The action of the quantum perceptron N over $|\psi_0\rangle$ is shown in Equation (23), where N calculates the output for all possible weights (or all neurons in

$$\begin{aligned} \mathsf{N}|\psi_0\rangle = \frac{1}{2}\mathsf{N}(|01\rangle|00\rangle|00\rangle|0\rangle|0\rangle + \\ |01\rangle|01\rangle|00\rangle|0\rangle|0\rangle + |01\rangle|10\rangle|00\rangle|0\rangle|0\rangle + \\ |01\rangle|11\rangle|00\rangle|0\rangle|0\rangle) = \\ \frac{1}{2}(\mathsf{N}|01\rangle|00\rangle|00\rangle|0\rangle|0\rangle + \mathsf{N}|01\rangle|01\rangle|00\rangle|0\rangle|0\rangle + \\ \mathsf{N}|01\rangle|10\rangle|00\rangle|0\rangle|0\rangle + \mathsf{N}|01\rangle|11\rangle|00\rangle|0\rangle|0\rangle) = \\ \frac{1}{2}(\mathsf{S}_{p_1p_2y}|01\rangle|00\rangle|0\odot 0, 1\odot 0\rangle|0\rangle|0\rangle + \\ \mathsf{S}_{p_1p_2y}|01\rangle|01\rangle|0\odot 0, 1\odot 1\rangle|0\rangle|0\rangle + \\ \mathsf{S}_{p_1p_2y}|01\rangle|10\rangle|0\odot 1, 1\odot 0\rangle|0\rangle|0\rangle + \\ \mathsf{S}_{p_1p_2y}|01\rangle|11\rangle|0\odot 1, 1\odot 1\rangle|0\rangle|0\rangle) = \\ |01\rangle|00\rangle|0,0\rangle|0\oplus 0\rangle|0\rangle + \\ |01\rangle|01\rangle|0,1\rangle|0\oplus 1\rangle|0\rangle + \\ |01\rangle|10\rangle|0,0\rangle|0\oplus 0\rangle|0\rangle + \\ |01\rangle|11\rangle|0,1\rangle|0\oplus 1\rangle|0\rangle) = \\ |01\rangle|00\rangle|0,0\rangle|0\rangle|0\rangle + |01\rangle|01\rangle|0,1\rangle|1\rangle|1\rangle + \\ |01\rangle|10\rangle|0,0\rangle|0\rangle|0\rangle + |01\rangle|11\rangle|0,1\rangle|1\rangle|1\rangle) \end{aligned} \tag{23}$$

5.4. *Neural network architecture*

We start this Section with an example of a classical multilayer perceptron and show an equivalent representation in a quantum computer. Let N be the neural network described in Fig. 1.

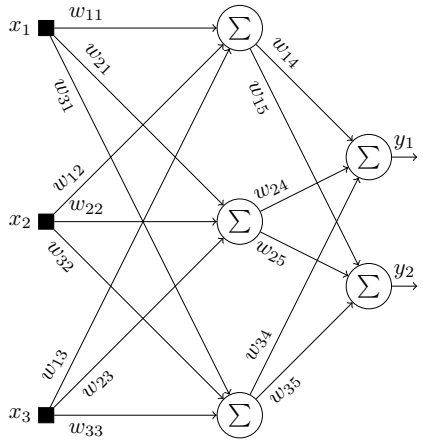


Fig. 1. Architecture of the multilayer quantum perceptron over a field.

The output of this network can be calculated as

$y = L_2 \cdot L_1 \cdot x$ using the three matrices L_1, L_2 and x described in Equation (24).

$$L_1 = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}, \quad L_2 = \begin{bmatrix} w_{14} & w_{24} & w_{34} \\ w_{15} & w_{25} & w_{35} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{24}$$

Weights, inputs and outputs in a classical neural network are real numbers. Here we suppose finite memory and we use elements of a finite field $(F, \oplus, \odot)$ to represent the neural network parameters. We can define a quantum operator $M_{3\times3,3\times1}$ that multiplies a 3×3 matrix with a 3×1 matrix. If $L_1 \cdot x = \begin{bmatrix} o_1 & o_2 & o_3 \end{bmatrix}^t$ we define the action of $M_{3\times3,3\times1}$ in Equation (25), where $w_i = w_{i1}, w_{i2}, w_{i3}$.

$$M_{3\times3,3\times1}|w_1, w_2, w_3, x_1, x_2, x_3, 0, 0, 0\rangle = |w_1, w_2, w_3, x_1, x_2, x_3, o_1, o_2, o_3\rangle \tag{25}$$

Each layer of the quantum perceptron over a field can be represented by an arbitrary matrix as in Equation (26),

$$M_{2\times3,3\times1} M_{3\times3,3\times1} |L_2\rangle |L_1\rangle |x\rangle |000\rangle |00\rangle \tag{26}$$

where $M_{3\times3,3\times1}$ acts on $|L_1\rangle$, $|x\rangle$ with output in register initialized with $|000\rangle$; and $M_{2\times3,3\times1}$ acts on $|L_2\rangle$, the output of the first operation, and the last quantum register. This matrix approach can be used to represent any feed-forward multilayer quantum perceptron over a field with any number of layers.

We suppose here that the training set and desired output are composed of classical data and that the data run forward. The supposition of classical desired output will allow us to superpose neural network configurations with its performance, as we will see in the next section.

5.5. *Learning algorithm*

In this Section, we present a learning algorithm that effectively uses quantum superposition to train a quantum perceptron over a field. Algorithms based on superposition have been proposed previously in Refs. 8, 27, 42. In these papers, a non-linear quantum operator proposed in Ref. 28, is used in the learning process. In Ref. 8 performances of neural networks in superposition are entangled with its representation. A non-linear algorithm is used to recover a neural network configuration with performance greater than a given threshold θ. A non-linear algorithm is used to recover the best neural network configuration. In Ref. 27 the nonlinear quantum operator is used in the learning process of a neurofuzzy network. In Ref. 42 a quantum associative neural network is proposed where a non-linear quantum circuit is used to increase the pattern recalling speed.

We propose a variant of the learning algorithm proposed in Ref. 8. The proposed quantum algorithm is named Superposition based Architecture Learning (SAL) algorithm. In the SAL algorithm the superposition of neural networks will store its performance entangled with its representation, as in Ref. 8. Later we will use a non-linear quantum operator to recover the architecture and weights of the neural network configuration with best performance.

In the classical computing paradigm, the idea of presenting an input pattern to all possible neural networks architectures is impracticable. To perform this idea classically one will need to create several copies of the neural network (one for each configuration and architecture) to receive all the inputs and compute in parallel the corresponding outputs. After calculating the output of each pattern for each neural network configuration, one can search the neural configuration with best performance. Yet classically the idea of SAL learning is presented in Fig. 2. For some neural network architectures, all the patterns in the training set $P = \{p_1, p_2, \cdots, p_k\}$ are presented to each of the neural network configurations. Outputs are calculated and then one can search the best neural network parameters.

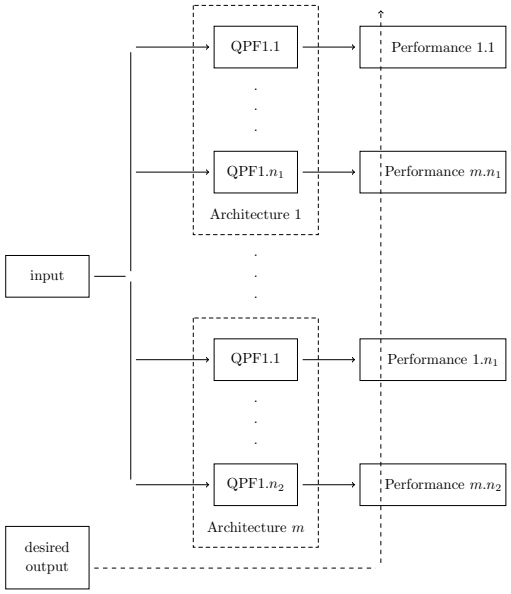


Fig. 2. Superposition based framework

In a quantum computer, the framework described in Fig. 2 can be implemented without the hardware limitations showed in the classical implementation. Let N_0, N_1, $\cdots$, N_{m-1} be m quantum operators representing neural networks with different architectures. A quantum circuit with quantum registers architecture selector $|a\rangle$ with $\lceil \log_2(m) \rceil$ qubits, input $|x\rangle$, weight $|w\rangle$ and output$|o\rangle$ can be created, where operator N_i is applied to $|x, w, o\rangle$ if and only if $|a\rangle = |i\rangle$. In Fig. 3 this approach is illustrated with $m = 4$.

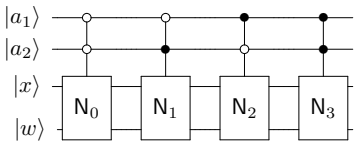


Fig. 3. Circuit to create a superposition with four neural network architectures.

If the qubits in quantum registers $|a\rangle$ and $|w\rangle$ are initialized with the quantum state $\mathsf{H}|0\rangle$, the circuit will be in a superposition representing all possible weights configuration for each architecture. Initializing the quantum register $|x\rangle$ with a pattern p, it is possible to present the pattern p to all neural network configurations in the superposition simultaneously.

Algorithm 2 is the SAL algorithm. SAL is a quantum-learning algorithm for any quantum neural network model in which input $|p\rangle$, output $|o\rangle$, weights $|w\rangle$, architecture selectors $|a\rangle$ and desired output $|d\rangle$ are represented in separated quantum registers. The main idea of SAL algorithm is to create a superposition of all possible neural network configurations in a finite set of architectures and apply a non-linear quantum operator to recover the architecture and weights of a neural network configuration with a desired performance.

The Algorithm 2 initialization is performed in Steps 1 to 6. Step 1 defines m quantum operators representing multi-layers QPF with different architectures. Steps 2 and 3 initialize all the weights and architecture selectors with the quantum state $H|0\rangle$. After this step we have all possible neural network configurations for the given architectures in superposition. In Steps 5 and 6 quantum registers performance and objective are initialized respectively, with the quantum states $|0\rangle_n$ and $|0\rangle$.

The for loop starting in Step 7 is repeated for each pattern p of the data set. Step 8 initializes quantum registers p, o and d respectively with a pattern $|p\rangle$, state $|0\rangle$ and its desired output $|d\rangle$. Step 9 presents the pattern to the neural networks, and the outputs are calculated. In this moment, the pattern is present to all neural networks configurations, because the weights and architecture selectors quantum registers are in a superposition of all possibles weights and architectures. In Steps 10 to 12, it is verified for each configuration in the superposition if the desired output $|d\rangle$ is equal to the calculated output $|o\rangle$. If they match, is added the value 1 for the performance quantum register. Step 13 is performed to allow the initialization of the next for loop.

$$|a\rangle|w\rangle|performance\rangle|objective\rangle = \sum_{\substack{w\in W,\\ a\in A}} |a\rangle|w\rangle|performance(w)\rangle|0\rangle \qquad (27)$$

After the execution of the for loop, the state of quantum registers weights w, architecture selectors a, *performance* and *objective* can be described as in Equation (27).

Algorithm 2 SAL

1: Let N_0, N_1, $\cdots$, N_m be quantum operators representing multi-layers QPF with diferent architectures.
2: Create a quantum circuit where the i-th network acts if and only if $|a\rangle = |i\rangle$.
3: Initialize all weights quantum registers with the quantum state $\mathsf{H}|0\rangle$.
4: Initialize all architecture quantum registers with quantum state $\mathsf{H}|0\rangle$.
5: Initialize a quantum register $|performance\rangle$ with the state $|0\rangle_n$.
6: Initialize a quantum register $|objective\rangle$ with the state $|0\rangle$.
7: **for each** pattern p and desired output d in the training set **do**
8: Initialize the register p, o , d with the quantum state $|p, 0, d\rangle$.
9: Calculate $\mathsf{N}|p\rangle$ to calculate network output in register $|o\rangle$.
10: **if** $|o\rangle = |d\rangle$ **then**
11: Add 1 to the register $|performance\rangle$
12: **end if**
13: Calculate $\mathsf{N}^{-1}|p\rangle$ to restore $|o\rangle$.
14: **end for**
15: Perform a non-linear quantum search to recover a neural network configuration and architecture with desired performance.

Steps 1 to 14 of Algorithm 2 can be performed using only linear quantum operators. In Step 15 a non-linear quantum operator NQ proposed in Ref. 28 will be used. Action of NQ is described in Equation (28) if at least one $|c_i\rangle$ is equal to $|1\rangle$ otherwise its action is described in Equation (29).

$$\mathsf{NQ}\left(\sum_i |\psi_i\rangle|c_i\rangle\right) = \left(\sum_i |\psi_i\rangle\right)|1\rangle \tag{28}$$

$$\mathsf{NQ}\left(\sum_i |\psi_i\rangle|c_i\rangle\right) = \left(\sum_i |\psi_i\rangle\right)|0\rangle \tag{29}$$

The non-linear search used in step 15 is described in Algorithm 3. The for loop in Step 1 of Algorithm 3 indicates that the actions need to be repeated for each quantum bit in the architecture and weights quantum registers. Steps 3 to 5 set the objective quantum register $|o\rangle$ to $|1\rangle$ if the performance quantum register p is greater than a given threshold θ. After this operation the state of quantum registers a, w and o can be described as in Equation (30).

$$\sum_{\substack{w\in(P(a,w)<\theta),\\ |b\rangle\neq|i\rangle}} |a\rangle|w\rangle|0\rangle + \sum_{\substack{w\in(P(a,w)\geq\theta),\\ |b\rangle=|i\rangle}} |a\rangle|w\rangle|1\rangle \tag{30}$$

Now that quantum register objective is set to 1 in the desired configurations, it is possible to perform a quantum search to increase the probability amplitude of the best configurations.

Algorithm 3 Non-linear quantum search

1: **for each** quantum bit $|b\rangle$ in quantum registers $|a\rangle|w\rangle$ **do**
2: **for** $i = 0$ **to** 1 **do**
3: **if** $|b\rangle = |i\rangle$ and $|p\rangle > \theta$ **then**
4: Set $|o\rangle$ to $|1\rangle$
5: **end if**
6: Apply NQ to $|o\rangle$
7: **if** $|o\rangle = |1\rangle$ **then**
8: Apply $\mathsf{X}^i \cdot \mathsf{NQ}$ to qubit $|b\rangle$
9: Apply X to $|o\rangle$
10: **end if**
11: **end for**
12: **end for**

Step 6 applies NQ to quantum register $|o\rangle$. If there is at least one configuration with $|b\rangle = |i\rangle$ then the action of NQ will set $|o\rangle$ to $|1\rangle$. In this case, Steps 7 to 10 set qubit $|b\rangle$ from a superposed state to the computational basis state $|i\rangle$.

Algorithm 3 performs an exhaustive non-linear quantum search in the architecture and weights space. If there is a neural network configuration with the desired performance in initial superposition, the search will return one of these configurations. Otherwise the algorithm does not change the initial superposition and the procedure can be repeated with another performance threshold.

The computational cost of Steps 1 and 2 of SAL is $O(m)$ and depends on the number of neural networks architectures. Steps 3 to 6 has computational cost $O(m + n_w)$, where n_w is the number of qubits to represent the weights. The for starting in Step 7 will be executed p times and each inner line has constant cost. Step 14 is detailed in Algorithm 3. Steps 3 to 9 of Algorithm 3 have constant computational cost and it will be repeated $2 \cdot (m + n_w)$ times. The

overall cost of the SAL algorithm is $O(p + m + n_w)$ where p is the number of patterns in the training set.

6. Discussion

Classical neural networks have limitations, such as i) the lack of an algorithm to determine optimal architectures, ii) memory capacity and iii) high cost learning algorithms. In this paper, we investigate how to use quantum computing to deal with limitation iii). To achieve this objective, we define a quantum neural network model named quantum perceptron over a field QPF and a nonlinear quantum learning algorithm that performs an exhaustive search in the space of weights and architectures.

We have shown that previous models of quantum perceptron cannot be viewed as a direct quantization of the classical perceptron.In other models of quantum neural networks weights and inputs are represented by a string of qubits, but the set of all possible inputs and weights with inner neuron operations does not form a field and there is no guarantee that they are well defined operations. To define QPF we propose quantum operators to perform addition and multiplication such that the qubits in a computational basis forms a field with these operations. QPF is the unique neuron with these properties. We claim that QPF can be viewed as a direct quantization of a classical perceptron, since when the qubits are in the computational basis the QPF acts exactly as a classical percetron. In this way, theoretical results obtained for the classical perceptron remains valid to QPF.

There is a lack of learning algorithms to find the optimal architecture of a neural network to solve a given problem. Methods for searching near optimal architecture use heuristics and perform local search in the space of architectures as *eg.* trough evolutionary algorithms or meta-learning. We propose an algorithm that solves this open problem using a nonlinear quantum search algorithm based on the learning algorithm of the NNQA. The proposed learning algorithm, named SAL, performs a non-linear exhaustive search in the space of architecture and weights and finds the best architecture (in a set of previously defined architectures) in linear time in relation to the number of patterns in the training set. SAL uses quantum superposition to allow initialization of all possible architectures and weights in a way that the architecture search is not biased by a choice of weights configuration. The desired architecture and weight configuration is obtained by the application of the nonlinear search algorithm and we can use the obtained neural network as a classical neural network. The QPF and SAL algorithm extend our theoretical knowledge in learning in quantum neural networks.

Quantum computing is still a theoretical possibility with no actual computer, an empirical evaluation of the QPF in real world problems is not yet possible, "quantum computing is far from actual application[8]". Studies necessary to investigate the generalization capabilities of SAL algorithm through a cross validation procedure cannot be accomplished with actual technology.

The simulation of the learning algorithm on a classical computer is also not possible due to the exponential growth of the memory required to represent quantum operators and quantum bits in a classical computer. Fig. 4 illustrates the relationship between the number of qubits and the size of memory used to represent a quantum operator.

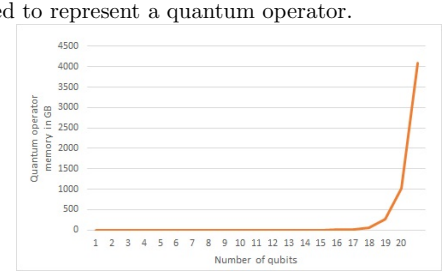


Fig. 4. Relation between the number of qubits n and size of quantum operators with matricial representation in $\mathbb{R}^{2^n \times 2^n}$ using standard C++ floating point precision

To illustrate the impossibility of carrying out an empirical analyses of Algorithm 2 let us consider the number of qubits necessary to represent a perceptron to learn the Iris dataset[43]. Let N be a quantum perceptron over a n-dimensional field F, then each attribute of the dataset and each weight of the network will be represented by n quantum bits. Iris database has 4 real entries, then the perceptron will have 4 weights. Excluding auxiliary quantum registers, weights and inputs will be represented by $8n$ quantum bits. An operator on 8n quantum bits will be represented by a matrix $2^{8n} \times 2^{8n}$. The number of bytes required to represent a $2^{8n} \times 2^{8n}$ real matrix using the standard C++ floating point data type is

$f(n) = 4 \cdot (2^{8n})^2$. Note that using only three quantum bits to represent the weights and input data the memory required for simulation is 1024 terabytes. Thus the (quantum or classical) simulation of the learning algorithm in real or synthetic problems is not possible with the current technology.

The multilayer QPF is a generalization of a multilayer classical perceptron and their generalization capabilities are at least equal. There is an increasing investment in quantum computing by several companies and research institutions to create a general-purpose quantum computer an it is necessary to be prepared to exploit quantum computing power to develop our knowledge in quantum algorithms and models.

If there are two or more architectures with desired performance in the training set, Algorithm 3 will choose the architecture represented (or addressed) by the string of qubits with more 0's. This information allows the use of SAL to select the minimal architecture that can learn the training set.

Nonlinear quantum mechanics have been study since the eighties[44] and several neural networks models and learning algorithms used nonlinear quantum computing[45; 8; 27; 42], but the physical realizability of nonlinear quantum computing is still controversial [28; 46]. A linear version of SAL needs investigation. The main difficulty is that before step 15 the total probability amplitude of desired configurations is exponentially smaller than the probability amplitude of undesired configurations. This is an open problem and it may be solved performing some iterations of "classical" learning in the states in the superposition before performing the recovery of the best architecture. Another possibility which we are investigating is to use Closed Timelike Curves[47; 48] known to make quantum computing have the power of the complexity class polynomial space (PSPACE).

7. Conclusion

We have analyzed some models of quantum perceptrons and verified that some of previously defined quantum neural network models in the literature does not respect the principles of quantum computing. Based on this analysis, we presented a quantum perceptron named quantum perceptron over a field (QPF). The QPF differs from previous models of quantum neural networks since it can be viewed as a direct generalization of the classical perceptron and can be trained by a quantum learning algorithm.

We have also defined the architecture of a multilayer QPF and a learning algorithm named Superposition based Architecture Learning algorithm (SAL) that performs a non-linear search in the neural network parameters and the architecture space simultaneously. SAL is based on previous learning algorithms. The main differences of our learning algorithm is the ability to perform a global search in the space of weights and architecture with linear cost in the number of patterns in the training set and in the number of bits used to represent the neural network. The principle of superposition and a nonlinear quantum operator are used to allow this speedup.

The final step of Algorithm 2 is a non-linear search in the architecture and weights space. In this step, free parameters will collapse to a basis state not in superposition. One possible future work is to analyze how one can use the neural network with weights in superposition. In this way one could take advantage of superposition in a trained neural network.

Acknowledgments

This work is supported by research grants from CNPq, CAPES and FACEPE (Brazilian research agencies). We would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

1. Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563–591, 1980.
2. R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467, 1982.
3. David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
4. L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, 1997.
5. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

6. Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Phys. Rev. A*, 58:915–928, Aug 1998.
7. A. Narayanan and T. Menneer. Quantum artificial neural networks architectures and components. *Information Sciences*, 128(3-4):231–255, 2000.
8. M. Panella and G. Martinelli. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications*, 39(1):61–77, 2011.
9. A. J. da Silva, Wilson R. de Oliveira, and T. B. Ludermir. Classical and superposed learning for quantum weightless neural networks. *Neurocomputing*, 75(1):52 – 60, 2012.
10. D. Ventura and T. Martinez. Quantum associative memory. *Information Sciences*, 124(1-4):273 – 296, 2000.
11. C. A. Trugenberger. Probabilistic quantum memories. *Phys. Rev. Lett.*, 87(6):067901, 2001.
12. Wei-Yen Hsu. Application of quantum-behaved particle swarm optimization to motor imagery eeg classification. *International journal of neural systems*, 23(6):1350026, 2013.
13. HB Duan, CF Xu, and Zhi-Hui Xing. A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. *International Journal of Neural Systems*, 20(1):39–50, 2010.
14. Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13(11):2567–2586, 2014.
15. J. Cabessa and H. T. Siegelmann. The super-turing computational power of plastic recurrent neural networks. *International Journal of Neural Systems*, 0(0):1450029, 0.
16. G. Zhang, H. Rong, F. Neri, and M. J. Pérez-Jiménez. An optimization spiking neural p system for approximately solving combinatorial optimization problems. *International Journal of Neural Systems*, 24(05):1440006, 2014.
17. M. González, D. Dominguez, F. B. Rodríguez, and Á. Sánchez. Retrieval of noisy fingerprint patterns using metric attractor networks. *International Journal of Neural Systems*, 24(07):1450025, 2014.
18. Akio Yamazaki and Teresa B. Ludermir. Neural network training with global optimization techniques. *International Journal of Neural Systems*, 13(02):77–86, 2003. PMID: 12923920.
19. Hamid Beigy and Mohammad R. Meybodi. Backpropagation algorithm adaptation parameters using learning automata. *International Journal of Neural Systems*, 11(03):219–228, 2001. PMID: 11574959.
20. S. C. Kak. On quantum neural computing. *Information Sciences*, 83(3):143–160, 1995.
21. N. Kouda, N. Matsui, H. Nishimura, and F. Peper. Qubit neural network and its learning efficiency. *Neural Comput. Appl.*, 14(2):114–121, 2005.
22. R. Zhou and Q. Ding. Quantum m-p neural network. *International Journal of Theoretical Physics*, 46:3209–3215, 2007.
23. Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. Quantum walks on graphs representing the firing patterns of a quantum neural network. *Physical Review A - Atomic, Molecular, and Optical Physics*, 89, 2014.
24. Michel A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
25. C. A. Trugenberger. Quantum pattern recognition. *Quantum Information Processing*, 1:471–493, 2002.
26. Leandro M. Almeida and Teresa B. Ludermir. A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks. *Neurocomputing*, 73(7–9):1438 – 1450, 2010. Advances in Computational Intelligence and Learning 17th European Symposium on Artificial Neural Networks 2009 17th European Symposium on Artificial Neural Networks 2009.
27. Massimo Panella and Giuseppe Martinelli. Neurofuzzy networks with nonlinear quantum learning. *Fuzzy Systems, IEEE Transactions on*, 17(3):698–710, 2009.
28. D. S. Abrams and S. Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for *np*-complete and *p* problems. *Phys. Rev. Lett.*, 81(18):3992–3995, 1998.
29. M. V. Altaisky. Quantum neural network. Technical report, Joint Institute for Nuclear Research, Russia, 2001.
30. B. Ricks and D. Ventura. Training a quantum neural network. In *Advances in Neural Information Processing Systems*. Cambridge, MA, 2004.
31. Hou Xuan. Research on quantum adaptive resonance theory neural network. In *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, volume 8, pages 3885–3888, Aug 2011.
32. Alaa Sagheer and Mohammed Zidan. Autonomous Quantum Perceptron Neural Network. *arXiv preprint arXiv:1312.4149*, pages 1–11, 2013.
33. Michael Siomau. A quantum model for autonomous learning automata. *Quantum Information Processing*, 2014.
34. E. C. Behrman, L. R. Nash, J. E. Steck, V. G. Chandrashekar, and S. R. Skinner. Simulations of quantum neural networks. *Information Sciences*, 128(3-4):257–269, 2000.
35. Rigui Zhou, Ling Qin, and Nan Jiang. Quantum perceptron network. In *Artificial Neural Networks–ICANN 2006*, volume 4131, pages 651–657, 2006.
36. P. Li, H. Xiao, F. Shang, X. Tong, X. Li, and M. Cao. A hybrid quantum-inspired neural networks with sequence inputs. *Neurocomputing*, 117:81 – 90, 2013.