Quantum Perceptron with Dynamic Internal Memory

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

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

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

I. Introduction

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

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

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

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

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

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

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

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

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

II. Quantum computation

A. Quantum bits

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

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

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

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

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

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

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

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

B. Quantum operators

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

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

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

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

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

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

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

C. Quantum circuit

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

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

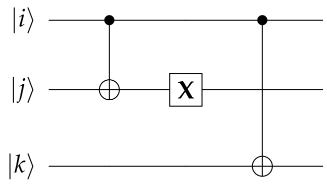


Fig. 1: An example of quantum circuit with two **CNOT** operators and a **X** operator.

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

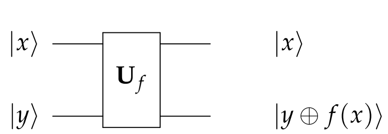


Fig. 2: Quantum circuit to represent an operation of an arbitrary boolean function f implemented by the quantum operator $\mathbf{U}_f$, where the symbol $\oplus$ is the XOR, the exclusive OR-operation (binary addition modulo 2). The result of the application of $|x\rangle$ in the function will be appear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

D. Quantum Fourier Transform

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

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

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

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

III. QUANTUM NEURONS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

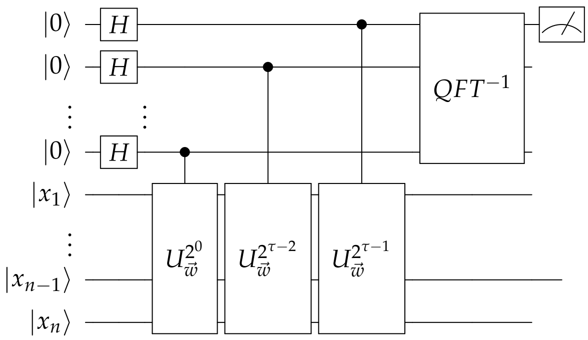


Fig. 3: Quantum perceptron circuit proposed in [17].

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

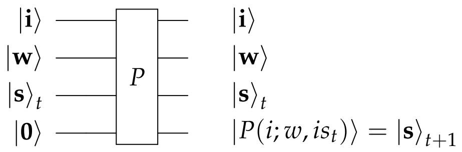


Fig. 4: Quantum circuit of the quantum perceptron of type A. This model considers that the internal state of the neuron is the neuron output in the last iteration.

IV. Quantum Perceptron Models with internal memory

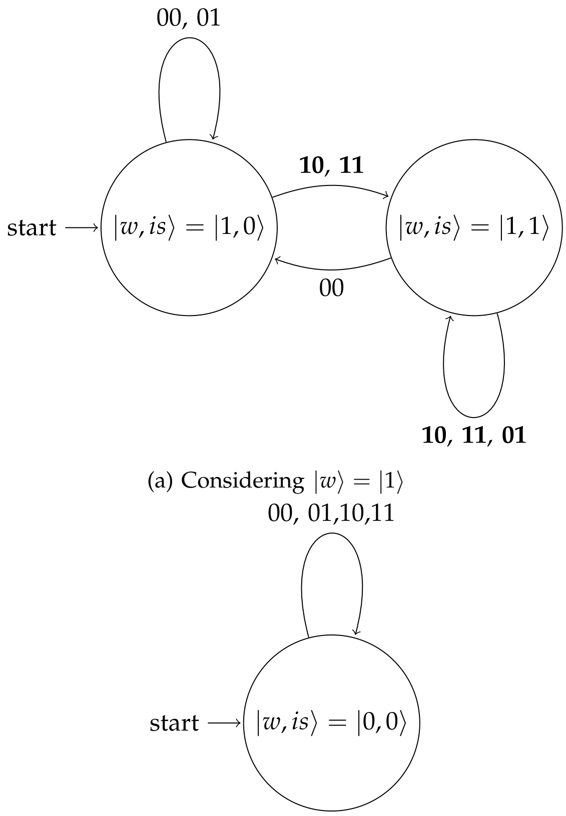
In this section, we propose to discuss the internal memory modification as a feature of a quantum perceptron. Then we modify the quantum perceptron proposed in [17] and we discuss how this modification affects the functioning of the neuron. The quantum perceptron of [17] is considered as a black box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

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

A. Quantum perceptron with internal memory type A

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

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



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

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

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

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

B. Quantum perceptron with internal memory type B

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

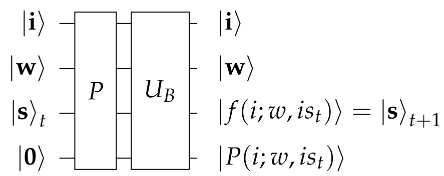


Fig. 6: Quantum circuit of the quantum perceptron of type B. This neuron model has the internal state modified by a reversible function in function of the input, weights and internal state in the previous iteration.

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

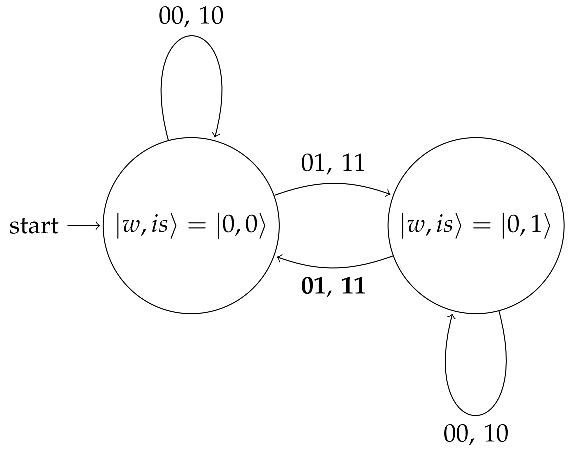
In this neuron model, one increases the amount of functions to be possible to alter the internal state. Not only the neuron activation function can be considered to alter it. Temporal information or specific configuration of inputs and weights can modify the internal memory state to some specific values.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

C. Perceptron with internal memory type C

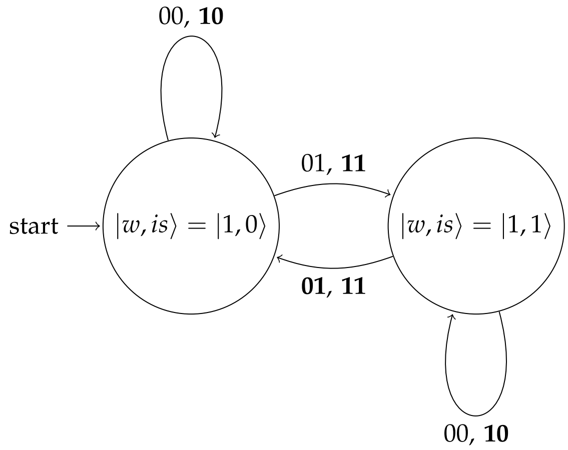
The quantum perceptron with internal memory which we name type C considers the possibility to modify the internal state by an arbitrary boolean function. This is possible because we are considering ancillary qubits to help in this computation. In Figure 8, the quantum circuit of this model is shown.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

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

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



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



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

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

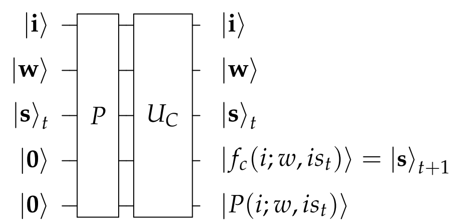
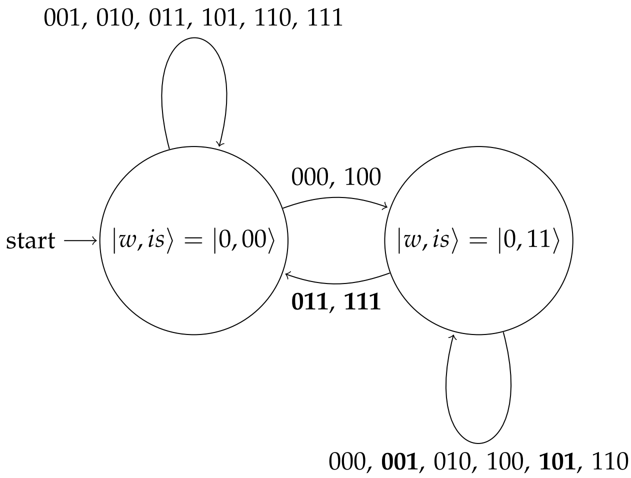
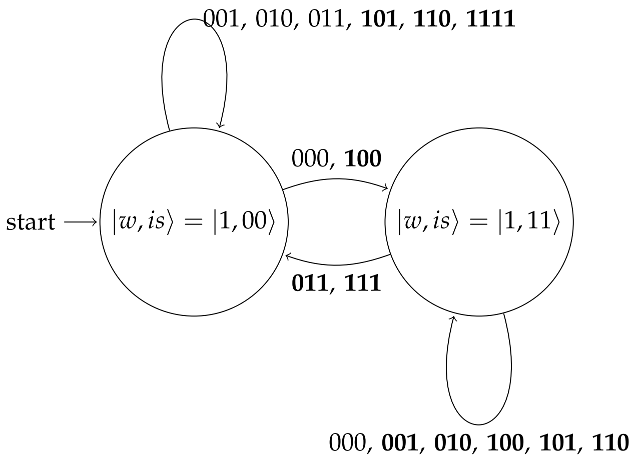


Fig. 8: Quantum circuit of the quantum perceptron of type C. This neuron mode has the internal state modified by an arbitrary boolean function.



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



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

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

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

V. Conclusion

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

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

Acknowledgment

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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