Physics Letters A 379 (2015) 660–663

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Simulating a perceptron on a quantum computer

Maria Schuld [a,*], Ilya Sinayskiy [a,b], Francesco Petruccione [a,b]

[a] Quantum Research Group, School of Chemistry and Physics, University of KwaZulu-Natal Durban, KwaZulu-Natal, 4001, South Africa
[b] National Institute for Theoretical Physics (NITheP), KwaZulu-Natal, 4001, South Africa

ARTICLE INFO

Article history:
Received 21 September 2014
Received in revised form 27 October 2014
Accepted 3 November 2014
Available online 24 December 2014
Communicated by A. Eisfeld

Keywords:
Quantum neural network
Quantum machine learning
Quantum computing
Linear classification

ABSTRACT

Perceptrons are the basic computational unit of artificial neural networks, as they model the activation mechanism of an output neuron due to incoming signals from its neighbours. As linear classifiers, they play an important role in the foundations of machine learning. In the context of the emerging field of quantum machine learning, several attempts have been made to develop a corresponding unit using quantum information theory. Based on the quantum phase estimation algorithm, this paper introduces a quantum perceptron model imitating the step-activation function of a classical perceptron. This scheme requires resources in $\mathcal{O}(n)$ (where n is the size of the input) and promises efficient applications for more complex structures such as trainable quantum neural networks.

1. Introduction

A perceptron is a mathematical model inspired by signal processing between neural cells that are assumed to be in either of the two states 'active' or 'resting'. It consists of n input nodes called *neurons* with values $x_k = \{-1, 1\}$, $k = 1, \dots, n$, that feed signals into a single output neuron y (Fig. 1 left). Each input neuron is connected to the output neuron with a certain strength denoted by a weight parameter $w_k \in [-1, 1)$ and the input–output relation is governed by the activation function

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

In other words, the net input $h(\vec{w}, \vec{x}) = \sum_{k=1}^{n} w_k x_k$ decides if the step-function activates the output neuron.[1] With their introduction by Rosenblatt in 1958 [1], perceptrons were a milestone in both the fields of neuroscience and artificial intelligence. Just like biological neural networks, perceptrons can learn an input–output function from examples by subsequently initialising $x_1, \dots, x_n$ with a number of example inputs, comparing the resulting outputs with the target outputs and adjusting the weights accordingly [2]. The high expectations of their potential for image classification tasks were disappointed when a study by Minsky and Papert in 1969 [3] revealed that perceptrons can only classify linearly separable functions, i.e. there has to be a hyperplane in phase space that divides the input vectors according to their respective output (Fig. 2). An example for an important non-separable function is the XOR function. The combination of several layers of perceptrons to artificial neural networks (also called multi-layer perceptrons, see Fig. 1 right) later in the 1980s elegantly overcame this shortfall, and neural networks are up to today an exciting field of research with growing applications in the IT industry.[2]

Since two decades, quantum information theory [5,6] offers a fruitful extension to computer science by investigating how quantum systems and their specific laws of nature can be exploited in order to process information efficiently [7,8]. Recent efforts investigate methods of artificial intelligence and machine learning from a quantum computational perspective, including the 'quest for a quantum neural network' [9]. Some approaches try to find a quantum equivalent for a perceptron, hoping to construct the building block for a more complex quantum neural network [10–12]. A relatively influential proposal to introduce a quantum perceptron is Altaisky's [10] direct translation of Eq. (1) into the formalism of quantum physics, namely $|y\rangle = \hat{F} \sum_{k=1}^{m} \hat{w}_k |x_k\rangle$, where the neurons $y, x_1, \dots, x_n$ are replaced by qubits $|y\rangle, |x_1\rangle, \dots, |x_n\rangle$ and the weights w_k become unitary operators $\hat{w}_k$. The step activation function is replaced by another unitary operator $\hat{F}$. Unfortunately, this proposal has not been extended to a full neural network model. A significant challenge is for example the learning procedure, since the suggested rule inspired by classical learning,

* Corresponding author. Tel.: +27 (0)31 260 8173.
E-mail address: schuld@ukzn.ac.za (M. Schuld).

[1] Another frequent class of perceptrons use values $x_k = [-1, 1]$, $k = 1, \dots, n$, and the logistic sigmoid activation function $y = \mathrm{sgm}(\sum_{k=1}^{m} w_k x_k)$.

[2] Consider for example the latest developments in Google's image recognition algorithms [4].

http://dx.doi.org/10.1016/j.physleta.2014.11.061

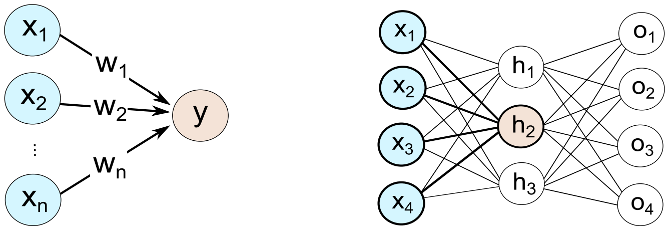

Fig. 1. (Colour online.) Left: Illustration of a perceptron model with input neurons $x_k = \{-1, 1\}$, weights $w_k \in [-1, 1)$, $k = 1, \dots, n$, and output neuron $y \in \{-1, 1\}$. Right: Perceptrons are the basic unit of artificial neural networks (here a feed-forward neural network). The network has an input layer, one hidden layer and an output layer, which get updated in chronological order. Every node or neuron computes its value according to the perceptron activation function Eq (1), so that the network maps an input $(x_1, \dots, x_4)$ to an output $(o_1, \dots, o_4)$.

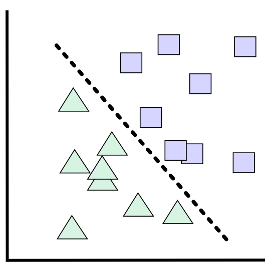

Fig. 2. A dataset is linearly separable if it can be divided regarding its outputs by a hyperplane in phase space.

$\hat{w}_k^{[t+1]} = \hat{w}_k^{[t]} + \eta(|d\rangle - |y^{[t]}\rangle)\langle x_k|$ with target output $|d\rangle$ and the learning rate $\eta \in [0, 1]$, does not maintain the unitarity condition for the operators $\hat{w}_k$. Other authors who pick up Altaisky's idea do not provide a solution to this severe violation of quantum theory [11,13,14] (or propose an according open quantum systems framework, in which the operators still have to remain completely positive and non-trace-increasing). Further models of quantum perceptrons can be found in the literature on quantum neural networks, but often remain vague in terms of the actual implementations [15,16], or do not apply quantum mechanics in a rigorous way [17,18]. An interesting exception is Elizabeth Behrman's work introducing a perceptron as the time evolution of a single quantum object [19], as well as Ricks and Ventura's ideas towards a superposition based learning procedure based on Grover's search algorithm [20].

This contribution introduces a unitary quantum circuit that with only a small number of extra resources simulates the nonlinear input–output function of a classical perceptron as given in Eq. (1). This quantum perceptron model has a high probability of reproducing the classical result upon measurement and can therefore be used as a classification device in quantum learning algorithms. The computational resources needed are comparable with the classical model, but the advantage lies in the fact that a quantum perceptron can process the entire learning set as a superposition, opening up new strategies for efficient learning. It can thus be seen as a building block of a more complex quantum neural network that harvests the advantages of quantum information processing.

2. The quantum perceptron algorithm

The quantum perceptron circuit is based on the idea of writing the normalised net input $\bar{h}(\vec{w}, \vec{x}) = \varphi \in [0, 1)$ into the phase of a quantum state $|x_1, \dots, x_n\rangle$, and applying the phase estimation algorithm with a precision of τ. This procedure will return a quantum state $|J_1, \dots, J_\tau\rangle$ which is the binary fraction representation of θ (or, equivalently, the binary integer representation of j in $\theta = \frac{j}{2^\tau}$), which is in turn a good approximation for φ. More

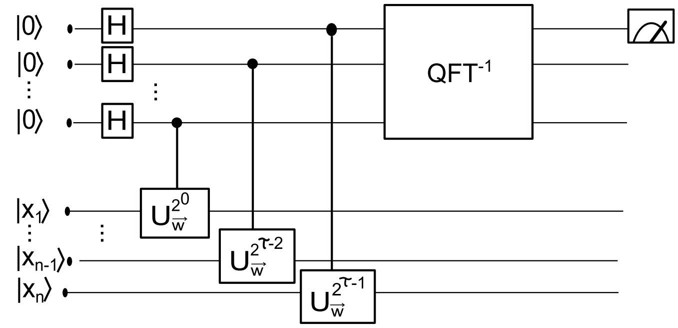

Fig. 3. Quantum circuit for the quantum perceptron model. See also [5].

precisely, the output encodes the phase via $\theta = J_1 \frac{1}{2} + \dots + J_\tau \frac{1}{2^\tau}$ (or $j = J_1 2^{\tau-1} + \dots + J_\tau 2^0$) [5]. The first digit of the output state of the quantum phase estimation algorithm, J_1, thus indicates if θ (and therefore with a good chance also φ) is bigger than $\frac{1}{2}$. The quantum perceptron consequently maps $(\vec{x}, \vec{w}) \to J_1$, which as we will see below reproduces the step activation function of a classical perceptron with a high probability.

To give a more detailed impression of the quantum perceptron circuit (see also Fig. 3), we assume an initial state $|0, \dots, 0\rangle|x_1, \dots, x_n\rangle = |0, \dots, 0\rangle|\psi_0\rangle$ composed of a register of τ qubits in state 0 as well as an input register $|\psi_0\rangle$ with n qubits encoding the binary states of the input neurons (note that in the quantum model, the -1 value is represented by a 0 state). Hadamard transformations on the τ zeroes in the first register lead to the superposition $\frac{1}{\sqrt{2^\tau}} \sum_{j=0}^{2^\tau-1} |J\rangle|x_1, \dots, x_n\rangle$, in which J is the binary representation of the integer j, and $|J\rangle = |J_1, \dots, J_\tau\rangle$. We apply an oracle $\mathcal{O}$ that writes j copies of a unitary transformation parameterised with the weights in front of the input register,

$$|J\rangle|\psi_0\rangle \xrightarrow{\mathcal{O}} |J\rangle U(\vec{w})^j|\psi_0\rangle. \qquad (2)$$

The unitary U writes the normalised input φ into the phase of the quantum state. This can be done using the decomposition into single qubit operators $U(\vec{w}) = U_n(w_n) \dots U_2(w_2)U_1(w_1)U_0$ with each

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

working on the input register's qubit x_k, and $\Delta\phi = \frac{1}{2n}$. U_0 adds a global phase of πi so that the resulting phase of state $|J\rangle|x_1, \dots, x_n\rangle$ is given by $\exp(2\pi i(\Delta\phi h(\vec{w}, \vec{x}) + 0.5) = \exp(2\pi i\varphi)$. For learning algorithms it might be useful to work with parameters represented in an additional register of qubits instead of parametrised unitaries, and below we will give an according variation of the quantum perceptron algorithm.

The next step is to apply the inverse quantum Fourier transform [21,5], QFT^{-1}, resulting in

$$\frac{1}{\sqrt{2^\tau}} \sum_{j=0}^{2^\tau-1} \exp^{2\pi i j\varphi} |J\rangle|\psi_0\rangle$$
$$\xrightarrow{\text{QFT}^{-1}} \sum_{j=0}^{2^\tau-1} \left(\frac{1}{2^\tau} \sum_{k=0}^{2^\tau-1} \exp^{2\pi i k(\varphi - \frac{j}{2^\tau})} \right) |J\rangle.$$

In case the phase can be exactly expressed as $\varphi = \frac{j}{2^\tau}$ for an integer j, the amplitude of all states except from $|J\rangle$ is zero and the algorithm simply results in $|J\rangle$. For cases $\varphi \neq \frac{j}{2^\tau}$, it can be shown that in order to obtain φ accurately up to m bits of precision with a success probability of $1 - \epsilon$, one has to choose $\tau = m + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$ [5]. Since we are only interested in the