ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Training a classical weightless neural network in a quantum computer

Adenilton J. da Silva[1,3] and Wilson R. de Oliveira[2] and Teresa B. Ludermir[1] *

1- Universidade Federal de Pernambuco - Centro de Informática
Cidade Universitária - 50740-560 - Recife/PE - Brazil

2- Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Dois Irmãos - CEP: 52171-900 - Recife/PE

Abstract. The purpose of this paper is to investigate a new quantum learning algorithm for classical weightless neural networks. The learning algorithm creates a superposition of all possible neural network configurations for a given architecture. The performance of the network over the training set is stored entangled with neural configuration and quantum search is performed to amplify the probability amplitude of the network with desired performance.

1 Introduction

Weightless neural networks [1] were proposed as pattern recognition systems. In general, learning algorithms for weightless neural networks are heuristics and there is no guarantee of convergence . Some algorithms perform the learning task with a single presentation of the training set. Here we denote these type of algorithms as single shot learning algorithms. For instance, it is possible to train a GSN [2] network with a single shot learning algorithm [3].

The objective of this paper is to present a quantum learning algorithm for classical weightless neural networks. The learning algorithm creates a superposition of all possible neural network configurations for a given architecture. The performance of the network over the training set is stored with neural configuration and quantum search is performed to amplify the probability amplitude of the network with desired performance.

Single shot learning algorithms provide fast learning capacities to Weightless Neural Networks (WNNs). However, the final neural network depends on presentation order of the patterns in the training set. In this paper, we propose a single shot learning algorithm for weightless neural networks that performs a global search in the space of parameters. It does not depend on pattern presentation order.

The remainder of this paper is divided into 4 sections. Section 2 presents the concept of weightless neural network. Section 3 presents concepts from quantum computation. Section 4 presents the main result of this work, the quantum single shot learning algorithm for WNNs. Finally, section 5 is the conclusion.

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

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

2 Weightless neural networks

The first weightless neuron proposed was the RAM neuron [1]. A RAM neuron is a self adaptive logic circuit. A n input RAM neuron has a memory C with 2^n memory positions. If the neuron receives a binary input x, then its output will be $C[x]$. The RAM neuron has probabilistic versions denoted Probabilistic Logic Neuron (PLN), Multivalued Probabilistic Logic Neuron (MPLN) and pRAM. In the PLN neuron it is possible to store 0, 1 and u in the memory positions. The u value corresponds to the probability of 50% to output 0 or 1. In the MPLN neuron there are a finite number of probabilities that can be stored in the memory position and in the pRAM neuron one can store a finite number of arbitrary precision probabilities.

A non-deterministic version of the PLN neuron is denoted GSN. The GSN neuron can receive inputs and can output values in the set $\{0, 1, u\}$. If one value u is in the input lines of the GSN, the neuron will access more than one memory position simultaneously. With input u the neuron will enter in more than one memory position. For instance, if a neuron with two inputs receives the signal $u0$, the neuron will access memory positions 00 and 10.

3 Quantum computation

For some problems there are quantum algorithms that solve problems faster than any known classical algorithm. For instance, the Grover's search algorithm [4], and the Shor's factoring algorithm [5] (that runs in polynomial time). In this Section, we present basic concepts of quantum computation necessary to understand this work.

A quantum bit is a vector in the complex vector space $\mathbb{C}^2$. The computational basis of $\mathbb{C}^2$ is the ordered set $\{|0\rangle, |1\rangle\}$, where $|0\rangle = [1, 0]^T$ and $|1\rangle = [0, 1]^T$. A quantum bit can be described as $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$. To represent more than one quantum bit it used the tensor product.

A quantum operator over n qubits is a unitary operator on a vector space with dimension 2^n. Operators in Equation 1 are examples of quantum operators over one qubit. For instance, $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$. With the quantum mechanical interpretation, one can think in state $|+\rangle$ as a superposition of the bits 0 and 1. The quantum bit is simultaneously in the bits 0 and 1.

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

Quantum parallelism uses states in superposition to calculate several values of a function in one single quantum step. Let $f : B^m \to B^n$ be a Boolean function. One can define the quantum operator U_f that implements the function f in a quantum computer, as described in Equation 2. If state $|x\rangle$ is a state in superposition and $|y\rangle = 0$, then U_f will calculate all the values of $f(x_i)$ for each x_i in the superposition as $U_f|x\rangle|0\rangle = \sum_i |x_i\rangle|f(x_i)\rangle$.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

$$U_f|x\rangle|y\rangle = |x\rangle|y + f(x)\rangle \tag{2}$$

Apparently one can use quantum parallelism to solve instantaneously any problem that can be represented as a Boolean function. But one cannot observe a quantum state directly. A measurement of a quantum bit $\alpha|0\rangle + \beta|1\rangle$ returns only $|0\rangle$ or $|1\rangle$ with probability $|\alpha|^2$ or $|\beta|^2$. After the measurement the state collapses to the measurement result. A quantum state with multiple qubits can be represented by $|\psi\rangle = \sum_i \alpha_i|i\rangle$. A measurement of this state will return $|i\rangle$ with probability $|\alpha_i|^2$. After the measurement the state collapses to $|\psi\rangle = |i\rangle$.

4 Single shot learning algorithm for WNNs

In [6] is presented a Nonlinear Single Shot Learning Algorithm (NSLA). The NSLA requires a single execution of the neural network, but a nonlinear quantum operator is used to train the network. Nonlinear quantum computation implies that P is equal to NP [7]. It is possible that nonlinear quantum computation breaks physics laws of the nature.

In this paper we propose a quantum linear algorithm to train weightless neural networks, denoted Linear quantum Single Shot learning Algorithm (LSSA). LSSA is based on the learning algorithm proposed in [8] and in the recovering algorithm of the quantum associative memory proposed in [9]. Before describing the LSSA we first show how to represent a WNN in a quantum computer. There are different ways to represent a neural network in a quantum computer. For instance, in [8] weights are qubits and in [10] weights are quantum operators.

Boolean weightless neural networks can be represented as a string of bits and its functionality can be realized as a Boolean function. For instance a RAM neuron with two inputs and 4 one-bit memory positions can be represented by a string of 6 bits. The first two bits represent the inputs and the 4 last ones represent the memory content.

Boolean functions can be directly generalized to quantum computation. So we represent the WNN in a quantum computer with a string of qubits. The action of the WNN will be represented by a quantum operator N. For example, a WNN with two inputs x_1, x_2 has its action in the computational basis represented by the operator N as described in Equation 3. The presentation of a pattern to a weightless neural network correspond to a multiplication of the neuron matrix representation and the pattern vector representation.

$$N|x_1x_2, p_1p_2p_3p_4, o\rangle = |x_1x_2, p_{00}p_{01}p_{10}p_{11}, o + p_{x_1x_2}\rangle \tag{3}$$

Algorithm 1 present the Linear quantum Single Shot learning Algorithm. In Step 1 the quantum operator N representing a classical weightless neural network is created. This operator can be represented by a matrix with dimension $2^n \times 2^n$, with $n = n_i + n_p + n_o$, where n_i is the number of inputs, n_p is the number of memory position, and n_o is the number of outputs.

In Step 2 all memory contents are initialized with the value $H|0\rangle$. With this initialization we have all possible neural networks for this configuration

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Algorithm 1: Linear quantum Single Shot learning Algorithm

1 Let N be a quantum operator representing a WNN. .
2 Initialize all memory registers with the quantum state $H|0\rangle$. .
3 Initialize a quantum register $|performance\rangle$ with the state $|0\rangle_n$.
4 Initialize a quantum register $|objective\rangle$ with the state $|0\rangle$.
5 **foreach** *pattern p and desired output d in the training set* **do**
6   Initialize the register p, o , d with the quantum state $|p, o, d\rangle$.
7   Present the pattern p to the network.
8   **if** $|o_i\rangle = |d_i\rangle$ **then**
9   Add 1 to the register $|performance\rangle$
10   **end**
11 **end**
12 For a given threshold θ, set $|objective\rangle = |1\rangle$ if $performance > \theta$.
13 Perform a quantum search to recover a state with the desired performance.

in superposition. Step 3 initializes a quantum state $|performance\rangle$ that will be used to store the performance of each neural network configuration in the superposition. Step 4 initializes a quantum register $|objective\rangle$ with the state $|0\rangle$. The quantum register $|objective\rangle$ will be used to perform a quantum search.

In Steps 5 to 11 patterns are presented to the network, and the quantum register performance is set to $|performance + 1\rangle$. At the conclusion of this loop each neural configuration in superposition will have its performance together with the neural network configuration.

In Step 12 the value of quantum register $|objective\rangle$ is set to $|1\rangle$ if its value is greater than a given threshold θ. In this moment each network configuration in the superposition will have $|objective\rangle = |1\rangle$ if the value in quantum register is greater than θ, otherwise $|objective\rangle = 0$.

4.1 Learning algorithm example

Here we show an instance of the learning algorithm execution to train a RAM neuron with two inputs to learn the two bit parity problem. This problem is represented by the training set $P = \{(|0,0\rangle, |0\rangle), |0,1\rangle, |1\rangle), |1,0\rangle, |1\rangle), |1,1\rangle, |0\rangle)\}$.

In steps 2 to 4 of the Algorithm is created a quantum state, where the parameters quantum register $|p\rangle$ is initialized with the state $H^{\otimes 4}|0\rangle$. Output quantum register is initialized in quantum state $|o\rangle = |0\rangle$ and performance quantum register is initialized with the quantum state $|performance\rangle = |0\rangle$. The resultant state is described in Equation 4.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

$$\frac{1}{4}\left[|input\rangle|memory\rangle|output\rangle|desiredoutput\rangle|performance\rangle|objective\rangle\right] = \frac{1}{4}\left[|input\rangle\left(\sum_{c\in\{0,1\}^4}|c\rangle\right)|0\rangle|desiredOutput\rangle|0\rangle|0\rangle\right] \quad (4)$$

In Steps 5 to 10, patterns of the training set are presented to the network and in each iteration the performance is updated. For instance, when the input $|01\rangle$ is fed to the network we initialize the quantum register *desiredOutput* with the quantum state $|1\rangle$.

$$\begin{aligned}
\frac{1}{4}N\left[|01\rangle\left(\sum_{c\in\{0,1\}^4}|c\rangle\right)|0\rangle|1\rangle|0\rangle|0\rangle\right] = \\
\frac{1}{4}N|01\rangle\,(|0000\rangle|0\rangle|1\rangle|0\rangle|0\rangle + |0001\rangle|0\rangle|1\rangle|0\rangle|0\rangle + |0010\rangle|0\rangle|1\rangle|0\rangle|0\rangle + \\
|0011\rangle|0\rangle|1\rangle|0\rangle|0\rangle + |\mathbf{0100}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + |\mathbf{0101}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + \\
|\mathbf{0110}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + |\mathbf{0111}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + |1000\rangle|0\rangle|1\rangle|0\rangle|0\rangle + \\
|1001\rangle|0\rangle|1\rangle|0\rangle|0\rangle + |1010\rangle|0\rangle|1\rangle|0\rangle|0\rangle + |1011\rangle|0\rangle|1\rangle|0\rangle|0\rangle + \\
|\mathbf{1100}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + |\mathbf{1101}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + |\mathbf{1110}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle + \\
|\mathbf{1111}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle)
\end{aligned} \quad (5)$$

The neuron will be applied in the state described in Equation 5. In the **boldfaced** configurations, the neuron output will be equal to the desired output. In this case $performance = performance + 1$. After the neuron execution we will have the quantum state described in Equation 6.

$$|01\rangle\left(\sum_{\substack{c\in\{0,1\}^4\\ c_1\neq 1}}|c\rangle\right)|0\rangle|1\rangle|0\rangle|0\rangle + |01\rangle\left(\sum_{\substack{c\in\{0,1\}^4\\ c_1=1}}|c\rangle\right)|1\rangle|1\rangle|1\rangle|0\rangle \quad (6)$$

In the next step we apply the quantum operator N^{-1}, the inverse of N, with the objective to restore the value of the calculated output to $|0\rangle$. Then one can present the next training pattern and desired output.

In next steps, the others patterns in the training set will be presented to the network. Only the neuron with memory configuration $|0110\rangle$ will match the desired output for all patterns. We use this information to set the quantum register $|objective\rangle = |1\rangle$ only for the configuration $|0110\rangle$.

Now we perform a quantum search in the neuron configuration space. The quantum register $|objective\rangle$ is used only in the Oracle application. The phase shift is applied only if $|objective\rangle = 1$. In this way, the configuration $|0110\rangle$ is returned with very high probability.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

5 Conclusion

The SSLA is the first learning algorithm for weightless neural networks that requires a single epoch and performs a global search in the parameter network space. The algorithm proposed is a quantum algorithm and its output is the set of parameters for a classical weightless neural network.

One possible future work is to simulate the learning process of a quantum weightless neural network. Simulation of quantum systems has exponential cost and it is possible only for very small problems. This simulation can be performed using a parallel model of computation, for instance using GPU processors [11].

Another possible future work is to develop a quantum weighted neural network with the properties found in quantum weightless neural networks. For instance with the capacity to simulate classical weighted neural networks [12]. This can be done creating a quantum neuron where inputs, weights, and outputs are elements of a field.

References

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

[2] E.C.D.B.C. Filho, M.C. Fairhist, and D. L. Bisset. Adaptive pattern recognition using the goal seaking neuron. *Pattern Recognition Letters*, 12:131–138, 1991.

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

[4] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997.

[5] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134, 1994.

[6] A.J. Silva, W.R. de Oliveira, and T.B. Ludermir. Single-shot learning algorithm for quantum weightless neural networks. In *Congresso Brasileiro de Inteligência Computacional*, 2013.

[7] Daniel S. Abrams and Seth Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for *np*-complete and *#p* problems. *Phys. Rev. Lett.*, 81:3992–3995, Nov 1998.

[8] M. Panella and G. Martinelli. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications*, 39:61—-77, july 2011.

[9] D. Ventura and T. Martinez. Quantum associative memory. *Information Sciences*, 128(1):273–296, 2000.

[10] M. V. Altaisky. Quantum neural network. Technical report, Joint Institute for Nuclear Research, Russia, 2001. Available online at the Quantum Physics repository:arXiv:quant-ph/0107012v2.

[11] Eladio Gutiérrez, Sergio Romero, María A. Trenas, and Emilio L. Zapata. Quantum computer simulation using the {CUDA} programming model. *Computer Physics Communications*, 181(2):283 – 300, 2010.

[12] W.R. de Oliveira, A.J. da Silva, T.B. Ludermir, A. Leonel, W.R. Galindo, and J.C.C. Pereira. Quantum logical neural networks. In *Neural Networks, 2008. SBRN '08. 10th Brazilian Symposium on*, pages 147–152, 2008.