Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 1509 – 1517

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Operations on Quantum Physical Artificial Neural Structures

Florian Neukart*, Sorin-Aurel Morar

University of Brasov, B-dul Eroilor nr.29, 500036 Brasov, Romania

Abstract

The advances that have been achieved in quantum computer science to date, slowly but steadily find their way into the field of artificial intelligence. Specifically the computational capacity given by quantum parallelism, resulting from the quantum linear superposition of quantum physical systems, as well as the entanglement of quantum bits seem to be promising for the implementation of quantum artificial neural networks. Within this elaboration, the required information processing from bit-level up to the computational neuroscience-level is explained in detail, based on the combined research in the fields of quantum physics and artificial neural systems.

Selection and peer-review under responsibility of DAAAM International Vienna

Keywords: Quantum Computer Science; Computational Neuroscience;

1. Introduction

From [1], where most of the introductory information has been taken from, a possible implementation of a quantum artificial neural network calculation, beginning with a superposition of all possible weight vectors has been proposed. The superposed weight vectors would allow classifying all training examples with respect to every weight vector at once. In the proposal, a performance register $|p\rangle$ is used to store the number of correctly classified training examples and updated continuously. The update of the performance register with respect to each configuration of the QANN creates an entanglement of $|p\rangle$and $|\psi\rangle$. Thus the oracle is

$$|p\rangle = i * o \tag{1}$$

wherei represents the number of training examples (inputs) and o the number of output neurons. As it may occur that either no configuration of a network within the superposition is able to classify all training examples correctly

* Corresponding author. Tel.: +49 171 12 199 12.
E-mail address: florian.neukart@edu.campus02.at

Selection and peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2014.03.148

(and therefore every vector has an equal chance of being measured) or the amount of time required for finding a vector is increasing with the number of bits in the weight vector and thus exponential complexity:

$$\mathrm{O}\left(\sqrt{2^{\mathrm{b}}/\mathrm{t}}\right) \tag{2}$$

For avoiding the first case, Ventura and Ricks suggest modifying the search oracle to

$$|p\rangle \geq i * o * p \tag{3}$$

wherep is the percentage of correctly classified training examples. With respect to quantum artificial neural networks, this means that any possible configuration of the quantum ANN is kept within the quantum linear superposition. However, there is still the problem of measurement, as measurement needs to be done when probability of receiving a desired result is high. Let assume a quantum register consisting of 64 Qbits each in the already known state

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

then every possible state, or every value (double) that can be expressed with these 64 bits may be measured, but with the same probability distribution, so any of these double values would exist in this quantum register at once. This is where quantum entanglement comes into play, as each possible weight vector is entangled with a slot in the performance register due to processing. Therefore, a measurement on the performance register when the probability of measuring the desired output is close to 1 will also reveal the desired overall weight vector (ANN configuration) due to the resulting loss of coherence in the processing bits. A more complicated substitute to eq. 3 might be an operator $\mathrm{U_f}$, applying the already known ANN performance calculations represented by a function f on $|\psi_i\rangle$:

$$x_{rmse} = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{5}$$

This could be done by summing the results from each $\mathrm{U_f}|\psi\rangle$ ($\mathrm{U_f}$ on each training set) into another quantum register $|\phi_i\rangle$, i representing the respective weight vector. When applying a quantum search on $|\phi\rangle$ after all, this then must include an average calculation, resulting in the overall RMSE. The structure of a quantum feed forward artificial neural network does not differ from a normal one, but the concept of linear superposition is one of the differences that can be described graphically [1] (

Fig.1. Quantum artificial neural network).

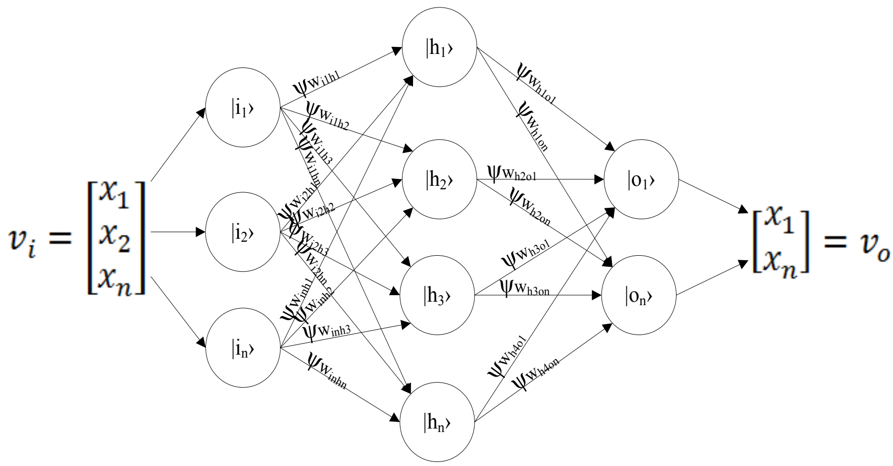

Fig.1. Quantum artificial neural network.

Fig.1. Quantum artificial neural network describes that neurons are represented as quantum registers, meaning that the input registers $|\mathrm{i}_1\rangle, \ldots, |\mathrm{i}_\mathrm{n}\rangle$ hold any possible input dataset of the training data. Assuming a QANN shall be

used to learn a problem based on input datasets each consisting of three attributes, it would contain three input registers (input neurons in a standard ANN), each holding any value its related training data attribute. The same holds for the hidden layer, represented by $|h_1\rangle, \dots, |h_n\rangle$, whose quantum neuron registers hold the calculated, weighted sum of the actual training input coming from $|i_1\rangle, \dots, |i_n\rangle$. The final layer $|o_1\rangle, \dots, |o_n\rangle$ holds the calculated output the neural network is capable of producing based on the actual training data set. Furthermore, the superpositions of the weight vectors are important to explain, as these hold every possible value they can represent, independent from the training data.

2. Quantum parallelism

Assuming a training set consisting of hundreds of input data sets each featuring a number x of attributes, then the input register would at first grow horizontally and secondly U_f in a classical computer would have to be applied on every single input data set consecutively not only once, but $n - 1$ times, where n represents the number of iterations (weigt adaptions) a training algorithm requires for adapting the overall weight vector of an ANN. Let further assume, 100 (fictive and not related to the inputs, but in numbers easier describe) Hadamard transformations (gates) would be applied on every Qbit before the application of U_f, like

$$U_f\left(H^{\otimes n} \otimes 1_m\right)\left(|0\rangle_n |0\rangle_m\right) = \frac{1}{2^{n/2}} \sum_{0 \le x < 2^n} U_f\left(|x\rangle_n |0\rangle_m\right) \tag{6}$$

then the final state would contain 2^{100} or $\approx 10^{30}$ applications of U_f (Mermin, 2007). However, quantum parallelism allows the performance of an exponentially high quantity of U_f in unitary time. This means that indeed every training set would require one application of U_f, but only once, as all possible weight vectors coexist in quantum linear superposition.

3. From basic operators to the quantum transfer function

From [1] we can further see that the whole processing of a quantum artificial neural network can be described as single transformation. However, it is required to detail the processing with regards to the activation or transfer function used by the neurons of the input- and hidden layer. Eq. 6 describes the sigmoid function as quantum function

$$|f(x)\rangle = \langle\phi|\psi\rangle = \frac{1}{1 + e^{-|x\rangle}} \tag{7}$$

where the quantum linear superposed $|x\rangle$ contains all possible values resulting from calculation based on the previous neuron layer output multiplied by the related weight vectors in superposition. This has not solved the basic challenge of arithmetic operations, like the multiplication or division required for all

$$V = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{8}$$

$$V|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \tag{9}$$

$$V|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ i \end{pmatrix} = i|1\rangle \tag{10}$$

After four subsequent applications of controlledV, the result is identity, and further three applications of the same operator result in its inverse, which is also its complex conjugate transpose $V^\dagger$. As mentioned beforehand, all operators must be linear, thus

$$V^\dagger V = V V^\dagger = I \tag{11}$$

where$V^\dagger$ is the complex conjugate transpose, or adjoint, of V, and I the identity operator. Thus, the resulting operator is c-$V^\dagger$. Back to cNOT, this operator can be built as in

Fig. 2. cNOT from H and V:

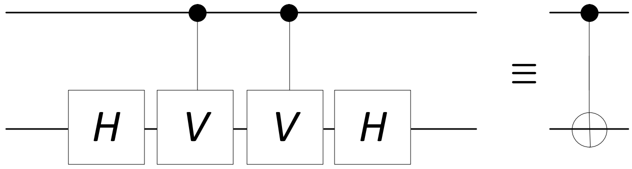

Fig. 2. cNOT from H and V.

$$|0\rangle|0\rangle \to |0\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \to |0\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \to |0\rangle|0\rangle \tag{12}$$

$$|0\rangle|1\rangle \to |0\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \to |0\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \to |0\rangle|1\rangle \tag{13}$$

$$|1\rangle|0\rangle \to |1\rangle\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}|1\rangle) \to |1\rangle\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}^2|1\rangle) \to |1\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \to |1\rangle|1\rangle \tag{14}$$

$$|1\rangle|1\rangle \to |1\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}|1\rangle) \to |1\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}^2|1\rangle) \to |1\rangle|0\rangle \tag{15}$$

This forms a very good basis for creating further operators, and indeed this needs to be done: for being able to perform all the arithmetic of the perceptron equations, another operator must be created, which is the ccNOT-operator (controlled-controlled-NOT), also known as Toffoli-gate [2] (
Fig. 3. Toffoli-gate with controlled V and
Fig. 4 . Toffoli-gate with complex conjugate transpose V):

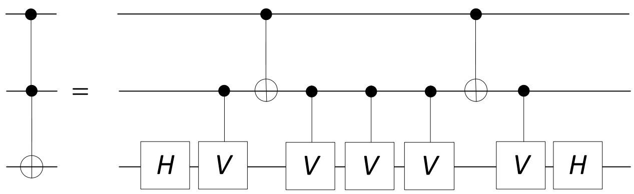

Fig. 3. Toffoli-gate with controlled V.

Or, with the already mentioned quantum gate $V^{\dagger}$:

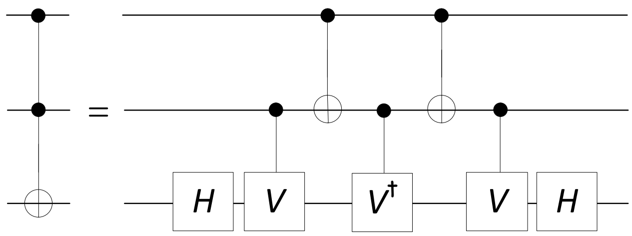

Fig. 4 . Toffoli-gate with complex conjugate transpose V.

This can be described in mathematical form by the quantum operations in the eqs. 16 and 17:

$$\begin{aligned}|110\rangle &\to |11\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \to |11\rangle\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}|1\rangle) \to |10\rangle\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}|1\rangle) \to |10\rangle\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}|1\rangle) \\ &\to |11\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \to |11\rangle\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}^2|1\rangle) = |11\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \to |111\rangle\end{aligned} \tag{16}$$

$$|111\rangle \rightarrow |11\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \rightarrow |11\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}|1\rangle) \rightarrow |10\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}|1\rangle) \rightarrow |10\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}|1\rangle) \rightarrow |11\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}|1\rangle) \rightarrow |11\rangle\frac{1}{\sqrt{2}}(|0\rangle - \mathrm{i}^2|1\rangle) = |11\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow |110\rangle \tag{17}$$

The Toffoli-gate, as controlled-controlled-NOT gate, features two control-bits, one more than the original cNOT-gate. The target bit is flipped only, and only when both of the control bits feature the state $|1\rangle|1\rangle$. Furthermore this provides a very important feature for multiplication (which in fact on binary level is an addition), namely a reversible AND-gate, if the target has initially featured the state $|0\rangle$: the target becomes the logical AND-operator of the two control bits as described in eq. 18 [3]:

$$|x_1, x_2\rangle|0\rangle \longmapsto |x_1, x_2\rangle|x_1 \wedge x_2\rangle \tag{18}$$

Thus, all required arithmetic operations, which are NOT, AND and XOR (cNOT), are available now as unitary transformations, which means that they can be stacked together to a larger unitary transformation for approaching the quantum perceptron equations. A combination of the Toffoli-gate and the cNOT-operator as quantum-adder form the basis for full addition and multiplication (
Fig. 5. Quantum addition):

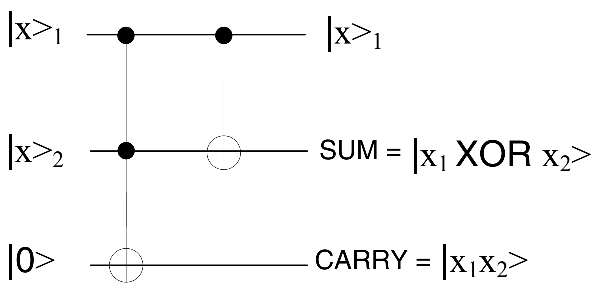

Fig. 5. Quantum addition.

Furthermore, any Boolean function which maps n bits from an input register to m bits of an output register may be evaluated (eq. 19).

$$\{0,1\}^n \rightarrow \{0,1\}^m \tag{19}$$

As the operation of AND is not reversible, it needs to be embedded into the ccNOT-operator. If the third bit has initially been set to 1 instead of 0 then the value $|x_1 \wedge x_2\rangle$ is flipped. Generally, the action of the Toffoli-gate is written as described in eq. 23:

$$|x_1, x_2\rangle|y\rangle \longmapsto |x_1, x_2\rangle|(y + (x_1 \wedge x_2)) \oplus 2^m\rangle \tag{20}$$

Summing up, the function evaluation may be described as unitary evolution of the input and output registers (eq. 24):

$$|x, y\rangle \longmapsto |x, (y + f(x)) \oplus 2^m\rangle \tag{21}$$

Thus, also more complex functions like the quantum sigmoid function (eq. 7), which include power functions like $e^{-|x\rangle}$ may be constructed, as the example of

$$f(x) = x^2 \tag{22}$$

A quantum network (the construct of quantum operators) calculating

$$f: \{0,1\}^2 \rightarrow \{0,1\}^3 \tag{23}$$

such that eq. 22 acts as described in eqs. 24 – 27:

$$|00\rangle|000\rangle \longmapsto |00\rangle|000\rangle \tag{24}$$

$$|01\rangle|000\rangle \longmapsto |01\rangle|001\rangle \tag{25}$$

$$|10|000\rangle \longmapsto |10|100\rangle \tag{26}$$

$$|11|000\rangle \longmapsto |11\rangle|001\rangle \tag{27}$$

which in terms of quantum computation is

$$|\mathrm{x}, 0\rangle \longmapsto |\mathrm{x}, \mathrm{x}^2 \oplus 8\rangle \tag{28}$$

an example being eq. 29, which is eq. 28 [3]:

$$3^2 \oplus 2^3 = 1 \tag{29}$$

It becomes obvious from the above explanations that arithmetic operations like multiplication are not ones that benefit from quantum effects, as these still are step-by-step-operations. The benefit occurs, when multiple of these operations for different configurations coexist and can be calculated simultaneously due to quantum parallelism in the quantum state of a physical system, like an artificial neural network in quantum linear superposition.

4. The desired configuration

As numerous configurations (towards infinity) configurations of the quantum artificial neural network's ray exist in linear superposition at once, it is tricky to figure out the correct one. At this point, Grover's algorithm [4] quantum database search algorithm, or more precisely, function inversion algorithm, used for searching a special configuration or item in an unsorted database (which is in case of the QANN the performance register in linear superposition) needs to be applied. Grover's algorithm provides the answer to when the system shall be measured, as measurement lets the superposition collapse, which eliminates all possible configurations of the QANN except the one measured. It is important to mention that the algorithm is capable of finding the desired solution in $\mathrm{O}(\sqrt{\mathrm{N}})$ time (iterations), which is nothing that can be done on a von Neumann computer.

5. Quantum artificial neural network configuration search function

When searching an unstructured database, containing $\mathrm{N} = 2^{\mathrm{n}}$ datasets, and with the search function f(x), called search oracle, then according to probability theory the probability P for finding the desired dataset x_{d} is k/N, where k is the number of randomly chosen database entries. Thus, on a von Neumann computer searching x_{d} requires an oracle querying all datasets (eq. 30).

$$O(N) = O(2^n) \tag{30}$$

calls of the oracle f(x), if

$$f(x) = \begin{cases} 1, & if x = x_d \\ 0, & if x != x_d \end{cases} \tag{31}$$

According to Grover's search algorithm, the number of oracle calls can be reduced dramatically, when inverting the phase of the desired basis states followed by an inversion of all basis states about the average amplitude of all states. The repetition of this process produces an increase of the amplitude of the desired basis state to near unity, followed by a corresponding decrease in the amplitude of the desired state back to its original magnitude [10]. Grover detected that this routine just needs to be called a number of repetitions that does not exceed $\frac{\pi}{4}\sqrt{\mathrm{N}}$, which is $\mathrm{O}(\sqrt{\mathrm{N}})$ iterations and thus, although the search is stochastic somehow, it outperforms a classical computer [1].When H_2 describes a two-dimensional Hilbert-space with the already known orthonormal basis of $\{|0\rangle, |1\rangle\}$ and $\{|0\rangle, |1\rangle, \dots, |\mathrm{N}-1\rangle\}$ describes the orthonormal basis the Hilbert space

$$H = \bigotimes_{0}^{n-1} H_2 \tag{32}$$

where⊗ again represents the tensor product. Here, the unitary transformation $\mathrm{U_f}$represents the oracle function f(x) by

$$|x\rangle \otimes |y\rangle \longrightarrow |x\rangle \otimes |f(x) \oplus y\rangle \tag{33}$$

$\oplus$again representing the exclusive or. However, as already indicated beforehand, the inner workings of the unitary transformation U_f are not known. However, it may be replaced by another computationally equivalent unitary transformation, namely

$$I_{|x_d\rangle}(|x\rangle) = (-1)^{f(x)}|x\rangle = \begin{cases} -|x_d\rangle, & if x = x_d \\ |x\rangle, & if x! = x_d \end{cases} \tag{34}$$

Equivalence is given, as

$$U_f \otimes |x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \left(I_{|x_d\rangle}(|x\rangle))\right) \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{35}$$

Furthermore, U_f results from a controlled $I_{|x_d\rangle}$ and two one-bit Hadamard transformations, and is, in fact, an inversion in H about the hyperplane orthogonal to $|x_d\rangle$. Thus, $I_{|x_d\rangle}$ may also be expressed as the tensor product of the desired ket with its bra and the identity transformation I, x_d now represented as a ray [5, 6]:

$$I_{|\psi\rangle} = I - 2|\psi\rangle\langle\psi| \tag{36}$$

6. Example processing

The already explained Hadamard-transformation needs to be applied for creating a superposition of all database entries, as described with eq. 6. In terms of the quantum artificial neural network, a conglomerate of all basis states each in a specific state represent one configuration. The Hadamard transformation applied to create $|\psi\rangle$ is

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} |c\rangle \tag{37}$$

whereN is the number of datasets and c represents the possible basis states of a neuron or even a whole artificial neural network. It is important to mention here that $|\psi\rangle$ contains $|x_d\rangle$ amongst all other basis states. Both states span a Euclidean plane, and Grover's algorithm rotates $|\psi\rangle$ about its origin as often as it takes this state to be as close to $|x_d\rangle$ as possible. This is the time the measurement needs to be done. However, before that the unitary transformation

$$Q = -I_{|\psi_d\rangle} I_{|x_d\rangle} \tag{38}$$

where$|\psi_d\rangle$ denotes the desired vector that is going to be measured, given from

$$Q = -HI_{|0\rangle}H^{-1}I_{|x_d\rangle} \tag{39}$$

$I_{|0\rangle}$, or $I_{|\psi_d\rangle}$ in its basic form, being an inversion with of the following character:

$$I_{|\psi\rangle} = I - 2|\psi\rangle\langle\psi| \tag{40}$$

From this, one can say that if $|\psi\rangle$ is a unit length ket in the Hilbert space H and U_f a unitary transformation on H, then

$$U_f I_{|\psi\rangle} U_f^{-1} = I_{U_f|\psi\rangle} \tag{41}$$

This means that U_f applied on the inversion of the ket $|\psi\rangle$, followed by the application of the inversion of the U_f, namely U_f^{-1}. It is important to know that the inversion of a unitary operator is always its adjunct. If this is not the case, the operator is not unitary. The result of equation (is then the inversion of $|\psi\rangle$ after U_f has worked on it. Back to Q and eq. 38 this means that Q is a rotation of a specific state $|\psi_d\rangle$ within $|\psi\rangle$ towards the desired value of the query $|x_d\rangle$ by a specific angle β. The rotation starts from two unit length vectors orthogonal to $|\psi_d\rangle$and $|x_d\rangle$, namely $|x_d^\perp\rangle$ and $|\psi_d^\perp\rangle$ with the same origin and β being the angle between these. The application of the transformation described in eq. 6 on these two vectors will result in a reflection in $|x_d^\perp\rangle$ followed by a reflection in $|\psi_d\rangle$, which is the same as a rotation by 2β. Summing up,

$$Q = -I_{|\psi_d^\perp\rangle} I_{|x_d\rangle} \tag{42}$$

is a rotation of $|\psi_d\rangle$ by 2β towards $|x_d\rangle$[5] (
Fig. 6. Rotation towards $|x_d\rangle$).

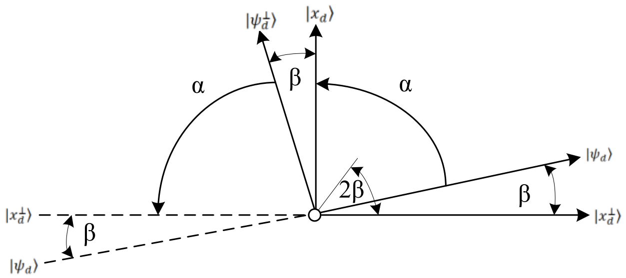

Fig. 6. Rotation towards $|x_d\rangle$.

Finally, the number of rotations needs to be determined, as too many rotations will rotate $|\psi_d\rangle$ past $|x_d\rangle$ and too few of them would lead to a stop way before it. Both are not desirable. From the latter explanations and Figure 45 - Rotation towards $|x_d\rangle$ it is obvious that

$$|\psi_d\rangle = \sin\beta|x_d\rangle + \cos\beta|x_d^\perp\rangle \tag{43}$$

After n rotations, or n applications of Q the resulting state is

$$|\psi_n\rangle = Q^n|\psi_d\rangle = \sin[(2n+1)\beta]|x_d\rangle + \cos[(2n+1)\beta]|x_d^\perp\rangle \tag{44}$$

Lomonaco (Lomonaco, 2000) describes moreover that the target now is to find an integer n so that $\sin[(2n+1)\beta]$ is as close to one as possible, or in another term, an integer that $(2n+1)\beta$ is very close to $\frac{\pi}{2}$. The angle α is complimentary to β:

$$\alpha + \beta = \frac{\pi}{2} \tag{45}$$

As a consequence,

$$n = \frac{\pi}{4\beta} - \frac{1}{2} = \left\lfloor \frac{\pi}{4\beta} \right\rfloor \tag{46}$$

Furthermore,

$$\frac{1}{\sqrt{N}} = \langle x_d|\psi_d\rangle = \cos\alpha = \cos\left(\frac{\pi}{2} - \beta\right) = \sin\beta \tag{47}$$

According to this,

$$\beta = \sin^{-1}\left(\frac{1}{\sqrt{N}}\right) \approx \frac{1}{\sqrt{N}} \tag{48}$$

and

$$n = \left\lfloor \frac{\pi}{4\sin^{-1}\left(\frac{1}{\sqrt{N}}\right)} \right\rfloor \approx \frac{\pi}{4}\sqrt{N} \tag{49}$$

Although the algorithm is generally described as database search algorithm, it would be more suitable to describe it as function inverter. This is, because for a given function f(x) the algorithm is able to determine y. Ricks and Ventura [7] made a very interesting proposal of how the optimal solution may be determined by a generalization of the of Grover's algorithm made of Boyer et al. [8].Another approach could be as follows: the output of a node would be f(x) and according to Grover's algorithm the determination of y is possible, which has to happen with a quantum search routine U_f. This search routine must then calculate backwards through the network, which is quite different from any other approach in neural network learning. Usually, y is given and one tries to determine f(x) and adapts f(x) through a learning algorithm as long as it is required to fulfil a stopping criterions, like the RMSE. However, as only f(x) is given, U_f is required to find the correct input to the desired output. Thus, the calculated output must be taken and the calculation must go backwards. Let assume, the perceptron equation is as follows:

$$|f(x)\rangle = tan\left(\sum_{i=1}^{n} |\omega\rangle_i \, |x\rangle_i + \theta\right) + \epsilon_t \tag{50}$$

Then U_f must be

$$|x\rangle_i = \frac{\arctan(|y\rangle_i - \epsilon_t) - \theta}{\sum_{i=1}^{n} |\omega\rangle_i} \tag{51}$$

and the error calculation

$$|y_{rmse}\rangle = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n} |y^2\rangle_i} \tag{52}$$

where n represents the number of outputs and y the input values.However, again and before that, the input register consisting of a number of n Qbits has to be put into superposition as it has already been done in eq. 54:

$$|\phi\rangle = H^{\otimes n}|0\rangle_n = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1} |x\rangle_n \tag{54}$$

Also, the already introduced unitary transformation V is required, plus an additional unitary transformation W acting on the input register as V with a fixed form not depending on a and preserving the component of any state along the standard (initial) state $|\phi\rangle$, but changing the sign of its component orthogonal to $|\phi\rangle$:

$$W = 2|\phi\rangle\langle\phi| - 1 \tag{55}$$

$|\phi\rangle\langle\phi|$ representing the projection operator on $|\phi\rangle$ [9]. Given these to transformations, Grover's algorithm applies the Product WV many times onto the input register in $|\phi\rangle$. Furthermore, each invocation of WV requires the quantum search routine or unitary operator U_f to be executed, which must be able to work with the superpositions of the QANN's states and which compares the entries of the database, or in the case of a quantum artificial neural network, the desired output with the calculated output. Summing up, in both algorithms the quantum search seeks to let the system fall into decoherence when the probability amplitudes for measuring the desired state near unity.

7. Conclusion

The work shows that all quantum operations required for processing a quantum artificial neural network can be constructed theoretically. Thus, an implementation of a quantum system capable of processing such a structure 'only' depends on a stable quantum computer.

References

[1] F. Neukart, Moraru Sorin-Aurel (2013): *On Quantum Computers and Artificial Neural Networks*. Journal of Signal Processing Research, vol. 2, 1, 2013.
[2] T. Toffoli (1981): *Mathematical Systems Theory* 14 13.
[3] Quantiki (2005): *Basic concepts in quantum computation* [2013-01-02]; URL: http://www.quantiki.org/wiki/Basic_concepts_in_quantum_computation
[4] L. K. Grover (1996): *A fast quantum mechanical algorithm for database search*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computation, pp.212-219.
[5] S. J. Lomonaco Jr. (2000): *Grover's Quantum Search Algorithm; Mathematics Subject Classification*. Primary 81P68; Secondary 81-01.
[6] A. Kitaev (1995): *Quantum measurements and the Abelian Stabilizer Problem*; L. D. Landau Institute for Theoretial Physics.
[7] B. Ricks, D. Ventura (2003): *Training a Quantum Neural Network*; Provo: Brigham Young University.
[8] M. Boyer et al. (1996): *Tight Bounds on Quantum Searching, Fourth Workshop on Physics and Computation*.
[9] D. N. Mermin (2007): *Quantum Computer Science: An Introduction*; Cambridge: Cambridge University Press.
[10] A. Ezhov, D. Ventura (-): *Quantum neural networks*, BSTU Laboratory of Artificial Neural Networks.