

# Artificial Associative Memory Using Quantum Processes

Dan Ventura

*fonix* corporation

180 West Election Road, Draper, UT 84020 USA

*dventura@fonix.com*

**Abstract** This paper discusses an approach to constructing an artificial quantum associative memory (QuAM). The QuAM makes use of two quantum computational algorithms, one for pattern storage and the other for pattern recall. The result is an exponential increase in the capacity of the memory when compared to traditional associative memories such as the Hopfield network. Further, the paper argues for considering pattern recall as a non-unitary process and demonstrates the utility of non-unitary operators for improving the pattern recall performance of the QuAM.

## 1. Introduction

The field of artificial neural networks (ANN) seeks, among other things, to develop algorithms for imitating in some sense the functionality of the brain. One particular area of interest is that of associative pattern recall, with perhaps the most well known approach being the Hopfield network [7]. Such ANN approaches to the pattern completion problem allow for associative pattern recall, but suffer severe storage restrictions. Storing patterns of length  $n$  requires a network of  $n$  neurons, and the number of patterns,  $m$ , is then limited by  $m \leq kn$ , where typically  $.15 \leq k \leq .5$ .

Quantum computation was introduced in the mid 1980s [4]. The field offers exciting possibilities -- perhaps the most notable to date being the discovery of quantum computational algorithms for computing discrete logarithms and prime factorization in polynomial time, two problems for which no known classical polynomial-time solutions exist [9]. These algorithms provide theoretical proof not only that interesting computation can be performed at the quantum level but also that it may in some cases have distinct advantages over its classical cousin.

As quantum computer technology continues to develop, ANN methods that are amenable to and take advantage of quantum mechanical properties will become possible. In particular, can quantum computation be applied to ANNs for problems such as associative memory? Recently, work has been done in the area of combining classical neural networks with ideas from the field of quantum mechanics [1] [8] [12].

This paper briefly discusses a quantum associative memory (QuAM) developed in [12] and offers an important extension to that work by appealing to non-unitary operators. This is not typical of quantum

algorithms, and arguments for the utility of the extension as well as for its justification are given.

## 2. Quantum Concepts

Quantum computation is based upon physical principles from the theory of quantum mechanics, which is in many ways counterintuitive. Yet it has provided us with perhaps the most accurate physical theory ever devised. The theory is well-established and is covered in its basic form by many textbooks (see for example [5]). Several ideas are briefly reviewed here.

*Linear superposition* is closely related to the familiar mathematical principle of linear combination of vectors. Quantum systems are described by a wave function  $\psi$  that exists in a Hilbert space. The Hilbert space has a set of states,  $|\phi_i\rangle$ , that form a basis, and the system is described by a quantum state,

$$|\psi\rangle = \sum_i c_i |\phi_i\rangle. \quad (1)$$

$|\psi\rangle$  is said to be in a linear superposition of the basis states  $|\phi_i\rangle$ , and in the general case, the coefficients  $c_i$  may be complex. Use is made here of the Dirac bracket notation, where the ket  $|\cdot\rangle$  is analogous to a column vector, and the bra  $\langle\cdot|$  is analogous to the complex conjugate transpose of the ket.

*Coherence* and *decoherence* are closely related to the idea of linear superposition. A quantum system is said to be coherent if it is in a linear superposition of its basis states. According to quantum mechanics, if a coherent system interacts in any way with its environment, the superposition is destroyed. This loss of coherence is called decoherence and is governed by the wave function  $\psi$ . The coefficients  $c_i$  are called probability amplitudes, and  $|c_i|^2$  gives the probability of  $|\psi\rangle$  collapsing into state  $|\phi_i\rangle$  if it decoheres. Note that the wave function  $\psi$  describes a real physical system that must collapse to exactly one basis state. Therefore, the probabilities governed by the amplitudes  $c_i$  must sum to unity. This necessary constraint is expressed as the unitarity condition

$$\sum_i |c_i|^2 = 1. \quad (2)$$

Consider, for example, a discrete physical variable called spin. The simplest spin system is a two-state system, called a spin-1/2 system, whose basis states are usually represented as  $|\uparrow\rangle$  (spin up) and  $|\downarrow\rangle$  (spin down). In this system the wave function  $\psi$  is a distribution over

two values and a coherent state  $|\psi\rangle$  is a linear superposition of  $|\uparrow\rangle$  and  $|\downarrow\rangle$ . One such state might be

$$|\psi\rangle = \frac{2}{\sqrt{5}}|\uparrow\rangle + \frac{1}{\sqrt{5}}|\downarrow\rangle. \quad (3)$$

As long as the system maintains its quantum coherence it cannot be said to be either spin up or spin down. It is in some sense both at once. When this system decoheres the result is, for example, the  $|\uparrow\rangle$  state with probability  $(2/\sqrt{5})^2 = 0.8$ .

A simple two-state quantum system, such as the spin-1/2 system just introduced, is used as the basic unit of quantum computation. Such a system is referred to as a quantum bit or *qubit*, and renaming the two states  $|0\rangle$  and  $|1\rangle$ , it is easy to see why this is so.

*Operators* on a Hilbert space describe how one wave function is changed into another. Here they will be denoted by a capital letter with a hat, such as  $\hat{A}$ , and they may be represented as matrices acting on vectors. Using operators, an eigenvalue equation can be written  $\hat{A}|\phi_i\rangle = a_i|\phi_i\rangle$ , where  $a_i$  is the eigenvalue. The solutions  $|\phi_i\rangle$  to such an equation are called eigenstates and can be used to construct the basis of a Hilbert space as discussed above. In the quantum formalism, all properties are represented as operators whose eigenstates are the basis for the Hilbert space associated with that property and whose eigenvalues are the quantum allowed values for that property. Operators in quantum mechanics must be linear and further, operators that describe the time evolution of a state must be unitary so that  $\hat{A}^\dagger\hat{A} = \hat{A}\hat{A}^\dagger = \hat{I}$ , where  $\hat{I}$  is the identity operator, and  $\hat{A}^\dagger$  is the complex conjugate transpose of  $\hat{A}$ .

*Interference* is a familiar wave phenomenon. Wave peaks that are in phase interfere constructively while those that are out of phase interfere destructively. This phenomenon is common to all kinds of wave mechanics from water waves to optics, and the well-known double slit experiment proves empirically that interference also applies to the probability waves of quantum mechanics.

## 2.1. Quantum Algorithms

The field of quantum computation offers exciting possibilities -- the most important quantum algorithms discovered to date all perform tasks for which there are no classical equivalents. For example, Deutsch's algorithm [3] is designed to solve the problem of identifying whether a binary function is constant (function values are either all 1 or all 0) or balanced (the function takes an equal number of 0 and 1 values). Deutsch's algorithm accomplishes the task in order  $O(1)$  time, while classical methods require  $O(2^n)$  time. Simon's algorithm [10] is constructed for finding the periodicity in a 2-1 binary function that is guaranteed to possess a periodic element. Here again an exponential

speedup is achieved; however, admittedly, both these algorithms have been designed for artificial, somewhat contrived problems as a proof of concept. Grover's algorithm [6], on the other hand, provides a method for searching an unordered quantum database in time  $O(\sqrt{2^n})$ , compared to the classical lower bound of  $O(2^n)$ . Here is a real-world problem for which quantum computation provides performance that is classically impossible. Finally, the most well-known and perhaps the most important quantum algorithm discovered so far is Shor's algorithm for prime factorization [9]. This algorithm finds the prime factors of very large numbers in polynomial time, while the best classical algorithms require exponential time.

## 3. Quantum Associative Memory

The QuAM is composed mainly of two key quantum computational algorithms, one for pattern storage and the other for pattern recall. In [12], both processes were considered evolutionary in nature and the algorithms for their implementation were developed using unitary operators. The two algorithms are briefly described here with references provided for further detail.

### 3.1 Storing Patterns

A quantum algorithm for constructing a coherent state over  $n$  qubits to represent a set of  $m$  patterns is presented in [11]. The algorithm is implemented using a polynomial number (in the length and number of patterns) of elementary operations on one, two, or three qubits. The key operator in this process is

$$\hat{S}^p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{p-1}{p}} & \frac{-1}{\sqrt{p}} \\ 0 & 0 & \frac{1}{\sqrt{p}} & \sqrt{\frac{p-1}{p}} \end{bmatrix}, \quad (4)$$

where  $m \leq p \leq 1$ . This is actually a set of operators that are conditional transforms -- there is a different  $\hat{S}^p$  operator associated with each pattern to be stored. The algorithm also makes use of various versions of some standard quantum computational operators such as the Controlled-Not and Fredkin gates.

Now given a set  $P$  of  $m$  binary patterns of length  $n$ , the quantum algorithm for storing the patterns requires a set of  $2n+1$  qubits, the first  $n$  of which actually store the patterns and can be thought of as  $n$  neurons in a quantum associative memory. The remaining  $n+1$  qubits are ancillary qubits used for bookkeeping and are restored to the state  $|\bar{0}\rangle$  after every storage iteration. Each iteration through the algorithm makes use of a different  $\hat{S}^p$  operator and results in another pattern being incorporated into the quantum

system. The result is a coherent superposition of states that corresponds to the patterns, with the amplitudes of the states in the superposition all being equal. The algorithm requires  $O(mn)$  steps to encode the  $m$  patterns as a quantum superposition over  $n$  quantum neurons. This is optimal in the sense that just reading each instance once cannot be done any faster than  $O(mn)$ .

### 3.2 Completing Patterns

Grover has developed an algorithm for finding one item in an unsorted database [6]. Classically, if there are  $2^n$  items in the database, this requires  $O(2^n)$  queries to the database. However, Grover has shown how to do this using quantum computation with only  $O(\sqrt{2^n})$  queries. In the quantum computational setting, finding the item in the database means measuring the system and having the system collapse to the basis state which corresponds to the item in the database for which we are searching. The basic idea of Grover's algorithm is to invert the phase of the desired basis state and then to invert all the basis states about the average amplitude of all the states. Repetition of this process produces an increase in 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. The process has a period of  $\frac{\pi}{4}\sqrt{2^n}$ , and thus after  $O(\sqrt{2^n})$  queries, the system may be observed in the desired state with near certainty. Define

$$\hat{I}_\phi = \text{identity matrix except for } i_{\phi\phi} = -1 \quad (5)$$

which inverts the phase of the basis state  $|\phi\rangle$ ,

$$\hat{W} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (6)$$

which is often called the Walsh transform, and

$$\hat{G} = -\hat{W}\hat{I}_0\hat{W} \quad (7)$$

which is the inversion about average.

Now to perform the quantum search on a database of size  $2^n$ , where  $n$  is the number of qubits, begin with the system in the  $|\bar{0}\rangle$  state and apply the  $\hat{W}$  operator. This initializes all the states to have the same amplitude. Finally, apply the operator sequence  $\hat{G}\hat{I}_\tau$ , where  $|\tau\rangle$  is the state being sought,  $\frac{\pi}{4}\sqrt{2^n}$  times and observe the system.

### 3.3 Combining the Algorithms

A quantum associative memory (QuAM) can now be constructed from the algorithms of Sections 3.1 and 3.2. Define  $\hat{P}$  as an operator that implements the algorithm for memorizing patterns described in Section 3.1. Then the operation of the QuAM can be described as follows. Memorizing a set of patterns is simply

$$|\psi\rangle = \hat{P}|\bar{0}\rangle, \quad (8)$$

with  $|\psi\rangle$  being a quantum superposition of basis states, one for each pattern. Now, suppose we know  $n - k$  bits of a pattern and wish to recall the entire pattern. We use a modification of Grover's algorithm to recall the pattern as

$$|\psi\rangle = \hat{G}\hat{I}_P\hat{G}\hat{I}_\tau|\psi\rangle \quad (9)$$

followed by

$$|\psi\rangle = \hat{G}\hat{I}_\tau|\psi\rangle \quad (10)$$

repeated  $O(\sqrt{2^n})$  times, with  $\hat{I}_P$  inverting the phases of all the states representing stored patterns and  $\hat{I}_\tau$  inverting the phases of those states matching the  $n - k$  known bits. (Since there are  $2^k$  states that will match the  $n - k$  bits, there will be  $2^k$  states that have their phases inverted by the  $\hat{I}_\tau$  operator.) Thus, with  $2n + 1$  neurons (qubits) the QuAM can store up to  $2^n$  patterns in  $O(mn)$  time and requires  $O(\sqrt{2^n})$  time to recall a pattern. This last bound is somewhat slower than desirable and may perhaps be improved with an alternative pattern recall mechanism.

## 4. Non-Unitary Pattern Completion

As mentioned earlier, evolutionary operators in quantum mechanics must be unitary, and thus far we have treated both the storage and recall segments of the QuAM as evolutionary processes. In fact, in the case of the storage algorithm, evolutionary processes are a necessity since the system must maintain a coherent superposition that represents the stored patterns. Fortunately, the necessity of using unitary operators has not precluded the discovery of an efficient mechanism for pattern storage.

On the other hand, requiring the recall mechanism to be evolutionary seems to limit the efficiency with which the recall may be accomplished. This is because in the general case, the pattern recall problem will be equivalent to Grover's search of a random database, and his algorithm has been shown to be within a constant factor of optimal for such problems [2]. However, non-unitary operators do exist in quantum mechanics and in nature. For example, any observation of a quantum system can be thought of as an operator that is neither evolutionary nor unitary. In fact, this non-evolutionary behavior of quantum systems is just as critical to our understanding of quantum mechanics as is their evolutionary behavior. Now, since the pattern recall mechanism in the QuAM requires the decoherence and collapse of the system, it can be argued that pattern recall may be a non-unitary process. In which case, the use of unitary operators becomes unnecessary and, in fact, may be detrimental to the operation of the QuAM. For example, unlike the case of pattern storage, using unitary operators for pattern recall seems to preclude efficient recall of the correct pattern. This unwanted

effect is due to the unitary evolution of the system, which introduces spurious states into the superposition -- states that had zero amplitude after the initial pattern storage acquire non-zero amplitude during the pattern recall. Since the decoherence and collapse of a quantum wave function is non-unitary and since pattern recall in the QuAM requires decoherence and collapse at some point, why not make explicit use of this non-unitarity, in the pattern recall phase?

With this motivation, we can define a new set of (non-unitary) operators for the pattern recall phase of the QuAM. Call these new recall operators  $\hat{R}$  and represent them as matrices  $r_{\phi\chi}$  indexed by column and row as the basis states of the system. Now define a string  $q$  over the alphabet  $\{0, 1, ?\}$ , with '?' representing "don't know". We can represent queries to the associative memory with different values for  $q$ . Each such query would be represented as a unique non-unitary operator,  $\hat{R}^q$  defined as

$$r_{\phi\chi} = \begin{cases} 1 & \text{if } \phi \equiv \chi \text{ and } h(\phi, q) \geq 1 \\ -1 & \text{if } h(\phi, q) > h(\chi, q) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where  $h(a,b)$  is a hamming function, and the character '?' matches anything.

The '1' entries in the matrix allow certain states the possibility of being the recalled state, and in fact, any state whose label matches  $q$  in even one character is included. The '-1' entries in the matrix cause destructive interference in such a way that a state with maximal  $h$  value is always chosen.

As a very simple example, consider the case of  $n = 2$  qubits and suppose that the QuAM has stored the patterns "01" and "11" so  $|\psi\rangle = 1/\sqrt{2}|01\rangle + 1/\sqrt{2}|11\rangle$ . To query the system with  $q = "11"$  requires the operator

$$\hat{R}^{11} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

and results in

$$\hat{R}^{11}|\psi\rangle = |11\rangle. \quad (13)$$

Notice that if the QuAM had not stored the pattern "11", for example if  $|\psi\rangle = 1/\sqrt{2}|00\rangle + 1/\sqrt{2}|10\rangle$ , then a pattern as close to the query as possible (in terms of hamming distance) would be chosen (in this case, "10").

Now, suppose that we have the query  $q = "?1"$ . The appropriate operator is

$$\hat{R}^{?1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

(there are no interference terms because all the entries in the matrix have equal  $h$  values). The result with the original  $|\psi\rangle = 1/\sqrt{2}|01\rangle + 1/\sqrt{2}|11\rangle$  is now

$$\hat{R}^{?1}|\psi\rangle = 1/\sqrt{2}|01\rangle + 1/\sqrt{2}|11\rangle \quad (15)$$

with either outcome equally likely. For each query, the application of a single operator suffices.

## 5. Concluding Comments

Quantum processes can be used to implement useful artificial neural network constructs. Here, an artificial quantum associative memory is described that has a capacity far superior to its classical analogs. In addition, it is conceivable to consider the pattern recall mechanism as a non-unitary quantum process. Initial discussion of what sort of non-unitary operator might be useful in this context has produced a significant improvement in recall time over the more traditional quantum computational approach of treating the recall as a system evolution (and thus as a unitary process).

## References

- [1] E. Behrman, J. Niemel, J. Steck and S. Skinner, "A Quantum Dot Neural Network", *Proc. of the Workshop on Physics of Computation*, pp. 22-24, 1996.
- [2] M. Boyer, G. Brassard, P. Høyer and A. Tapp, "Tight Bounds on Quantum Searching", *Proc. of the Workshop on Physics of Computation*, pp. 36-43, 1996.
- [3] D. Deutsch and R. Jozsa, "Rapid Solution of Problems by Quantum Computation", *Proceedings of the Royal Society, London A*, v439, pp. 553-558, 1992.
- [4] D. Deutsch, "Quantum Theory, The Church-Turing Principle and the Universal Quantum Computer", *Proceedings of the Royal Society, London A*, v400, pp. 97-117, 1985.
- [5] R. Feynman, R. Leighton and M. Sands, *The Feynman Lectures on Physics*, v3, Addison-Wesley, 1965.
- [6] L. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", *Proceedings of the ACM Symposium on the Theory of Computing*, pp. 212-19, 1996.
- [7] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the National Academy of Scientists*, v79, pp. 2554-2558, 1982.
- [8] M. Perus, "Neuro-Quantum Parallelism in Brain-Mind and Computers", *Informatica*, v20, pp. 173-183, 1996.
- [9] P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", *SIAM Journal of Computing*, v26, pp. 1484-1509, 1997.
- [10] D. Simon, "On the Power of Quantum Computation", *SIAM Journal of Computation*, v26, pp. 1474-83, 1997.
- [11] D. Ventura and T. Martinez, "Initializing the Amplitude Distribution of a Quantum State", submitted to *Physical Review Letters*, June 16, 1998.
- [12] D. Ventura and T. Martinez, "Quantum Associative Memory", submitted to the *IEEE Transactions on Neural Networks*, June 16 1998.