A Quantum Associative Memory Based on Grover's Algorithm

Dan Ventura and Tony Martinez
Neural Networks and Machine Learning Laboratory (http://axon.cs.byu.edu)
Department of Computer Science
Brigham Young University, Provo, UT 84602 USA

Abstract

Quantum computation uses microscopic quantum level effects to perform computational tasks and has produced results that in some cases are exponentially faster than their classical counterparts. The unique characteristics of quantum theory may also be used to create a quantum associative memory with a capacity exponential in the number of neurons. This paper combines two quantum computational algorithms to produce a quantum associative memory. The result is an exponential increase in the capacity of the memory when compared to traditional associative memories such as the Hopfield network. The paper covers necessary high-level quantum mechanical ideas and introduces a quantum associative memory, a small version of which should be physically realizable in the near future.

1 Introduction

Assume a set $\wp$ of m binary patterns of length n. We consider the problem of pattern completion -- learning to produce one of the full patterns when presented with only a partial pattern. The trivial solution is simply to store the set of patterns as a lookup table or RAM. There are two reasons why this is not always the best solution. First, it requires that a unique address be associated with and remembered for each pattern. Second, the lookup table requires mn bits in order to store all the patterns. It is often desirable to be able to recall the patterns in an associative fashion, thus eliminating the need for explicit addressing. That is, given a partial pattern one would like to be able to "fill in" a reasonable guess as to the rest of the pattern. This may also be considered a form of generalization as the partial pattern may never have been seen during the learning of the pattern set $\wp$. Further, it would be beneficial if a smaller representation was possible.

To this end, various classical associative memory schemes have been proposed, perhaps the most well known being the Hopfield network [10]. Another well-known example is the bidirectional associative memory (BAM) [11]. These neural 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 $0.15 \leq k \leq 0.5$. This paper offers improvement by proposing a quantum associative memory that maintains the ability to recall patterns associatively while offering a storage capacity of $O(2^n)$ using only n neurons.

The field of quantum computation, which applies ideas from quantum mechanics to the study of computation, was introduced in the mid 1980's [6]. For a readable introduction to quantum computation see [1]. The field is still in its infancy and very theoretical but offers exciting possibilities for the field of computer science -- 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 [13]. 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.

Artificial neural networks (ANN) seek to provide ways for classical computers to learn rather than to be programmed. If quantum computers become a reality, then artificial neural network methods that are amenable to and take advantage of quantum mechanical properties will become possible. In particular, can quantum mechanical properties be applied to ANNs for problems such as associative memory? Recently, work has been done in the area of combining classical artificial associative memory with ideas from the field of quantum mechanics. Perus details several interesting mathematical analogies between quantum theory and neural network theory [12]. [15] goes a step further by proposing an actual model for a quantum associative memory. The work here further develops this model by exhibiting a physically realizable quantum system for acting as an associative memory.

This paper presents a unique reformulation of the pattern completion problem into the language of wave functions and operators. This reformulation may be generalized to a large class of computational learning problems, opening up the possibility of employing the capabilities of quantum computational systems for the solution of computational learning problems. Section

2 introduces some important ideas from quantum mechanics and briefly discusses quantum computation along with some of its early successes. Section 3 goes into more detail on one particular algorithm for quantum database search due to Grover [9]. Section 4 briefly describes a modification of the quantum algorithm, detailed elsewhere [16] for initializing a quantum system to represent a set of patterns, and the two algorithms are combined in section 5 to produce the quantum associative memory. Section 6 concludes and provides directions for further research.

2 Quantum Computation

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 (in terms of predicting experimental results) ever devised by science. The theory is well-established and is covered in its basic form by many textbooks (see for example [7]). Several necessary ideas that form the basis for the study of quantum computation 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 ψ 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 $|\psi\rangle$,

$$|\psi\rangle = \sum_i c_i|\phi_i\rangle. \tag{1}$$

$|\psi\rangle$ 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. In quantum mechanics the Hilbert space and its basis have a physical interpretation, and this leads directly to perhaps the most counterintuitive aspect of the theory. The counter intuition is this -- at the microscopic or quantum level, the state of the system is described by the wave function ψ, that is, as a linear superposition of all basis states (i.e. in some sense the system is in all basis states at once). However, at the macroscopic or classical level the system can be in only a single basis state. For example, at the quantum level an electron can exist in a superposition of many different energy levels; however, in the classical realm this cannot be.

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. A result of quantum mechanics is that if a system that is in a linear superposition of states 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 ψ. 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 ψ 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 constraint is expressed as the unitarity condition

$$\sum_i |c_i|^2 = 1. \tag{2}$$

In the Dirac notation, the probability that a quantum state $|\psi\rangle$ will collapse into an eigenstate $|\phi_i\rangle$ is written $|\langle\phi_i|\psi\rangle|^2$ and is analogous to the dot product (projection) of two vectors. 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 simple system the wave function ψ 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

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

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. Classically, it must be one or the other, and when this system decoheres the result is, for example, the $|\uparrow\rangle$ state with probability

$$|\langle\uparrow|\psi\rangle|^2 = \left(\frac{2}{\sqrt{5}}\right)^2 = 0.8. \tag{4}$$

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. It is important to note that operators in quantum mechanics must be linear operators and further that they 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 (magnify each other's amplitude) while those that are out of phase interfere destructively (decrease or eliminate each other's amplitude). This is a phenomenon common to all kinds of wave mechanics from water waves to optics. The well-known double slit experiment demonstrates empirically that interference also applies to the probability waves of quantum mechanics.

Entanglement is the potential for quantum states to exhibit correlations that cannot be accounted for classically. From a computational standpoint, entanglement seems intuitive enough -- it is simply the fact that correlations can exist between different qubits -- for example if one qubit is in the $|1\rangle$ state, another will be in the $|1\rangle$ state. However, from a physical standpoint, entanglement is little understood. The questions of what exactly it is and how it works are still not resolved. What makes it so powerful (and so little understood) is the fact that since quantum states exist as superpositions, these correlations somehow exist in superposition as well. When the superposition is destroyed, the proper correlation is somehow communicated between the qubits, and it is this "communication" that is the crux of entanglement and the key to quantum computation. It follows that while interference is a quantum property that has a classical cousin, entanglement is a completely quantum phenomenon for which there is no classical analog.

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 [5] 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 $O(n)$ time, while classical methods require $O(2^n)$ time. Simon's algorithm [14] 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 [9], on the other hand, provides a method for searching an unordered quantum database in time $O(\sqrt{N})$, compared to the classical lower bound of $O(N)$. Here is a real-world problem for which quantum computation provides performance that is classically impossible (though the speedup is less dramatic than exponential). Finally, the most well-known and perhaps the most important quantum algorithm discovered so far is Shor's algorithm for prime factorization [13]. This algorithm finds the prime factors of very large numbers in polynomial time, while the best classical algorithms require exponential time. Obviously, the implications for the field of cryptography are profound.

3 Grover's Algorithm

Lov Grover has developed an algorithm for finding one item in an unsorted database, similar to finding the name that matches a telephone number in a telephone book. Classically, if there are N items in the database, this would require on average $O(N)$ queries to the database. However, Grover has shown how to do this using quantum computation with only $O(\sqrt{N})$ queries. In the quantum computational setting, finding the item in the database means measuring the system and having the system collapse with near certainty to the desired 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 (for more details see [9] [8]). 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 is cyclical with a period of $\frac{\pi}{4}\sqrt{N}$, and thus after $O(\sqrt{N})$ queries, the system may be observed in the desired state with near certainty (with probability at least $1-\frac{1}{N}$). Interestingly this implies that the larger the database, the greater the certainty of finding the desired state [3]. Of course, if greater certainty is required, the system may be sampled k times boosting the certainty of finding the desired state to $1-\frac{1}{N^k}$. Define the following operators.

$$\hat{I}_\phi = \text{identity matrix except for } \phi\phi = -1, \tag{5}$$

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

$$\hat{W} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{6}$$

which is often called the Walsh or Hadamard transform. This operator, when applied to a set of qubits, performs a special case of the discrete fourier transform.

Now to perform the quantum search on a database of size $N = 2^n$, 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. Next apply the operator sequence $\hat{G}\hat{I}_\tau$ $\frac{\pi}{4}\sqrt{N}$ times, where $\hat{G} = -\hat{W}\hat{I}_{\bar{0}}\hat{W}$ can be thought of as rotating all the states about their average amplitude and $|\tau\rangle$ is the state being sought (and recall that operators are applied right to left). Finally, observe the system. This algorithm will be used to associatively "fill in" a pattern by finding a basis state that corresponds to the partial pattern to be completed.

4 Initializing the Quantum State

In [16] we presented a polynomial-time quantum algorithm for constructing a quantum state over a set of qubits to represent the information in a training set. The algorithm is implemented using a polynomial number (in the length and number of patterns) of elementary operations on one, two, or three qubits. Here the necessary operators are presented briefly and the reader is referred to [16] for details.

$$\hat{S}^p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{p-1}{p}} & \frac{-1}{\sqrt{p}} \\ 0 & 0 & \frac{1}{\sqrt{p}} & \sqrt{\frac{p-1}{p}} \end{bmatrix}, \qquad (7)$$

where $1 \le p \le m$. These operators form a set of conditional transforms that will be used to incorporate the set of patterns into a coherent quantum state. There will be a different $\hat{S}^p$ operator associated with each pattern to be stored. The interested reader may note that this definition of the $\hat{S}^p$ operator is slightly different than the original. This is because in this context, we are considering pattern memorization rather than pattern classification and therefore have no output class *per se*. Thus the phase of the coefficients becomes unimportant in this case.

$$\hat{F}^0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (8)$$

conditionally flips the second qubit if the first qubit is in the $|0\rangle$ state; $\hat{F}^1$ conditionally flips the second qubit if the first qubit is in the $|1\rangle$ state ($\hat{F}^1$ is the same as $\hat{F}^0$ except that the off-diagonal elements occur in the bottom right quadrant rather than in the top left). These operators are referred to elsewhere as Control-NOT because a logical NOT (state flip) is performed on the second qubit depending upon (or controlled by) the state of the first qubit.

$$\hat{A}^{00} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (9)$$

conditionally flips the third bit if and only if the first two are in the state $|00\rangle$. Note that this operator can be thought of as performing a logical AND of the negation of the first two bits, writing a 1 in the third if and only if the first two are both 0. Three other operators, $\hat{A}^{01}$, $\hat{A}^{10}$ and $\hat{A}^{11}$, are variations of $\hat{A}^{00}$ in which the off diagonal elements occur in the other three possible locations along the main diagonal. $\hat{A}^{01}$ can be thought of as performing a logical AND of the first bit and the negation of the second, and so forth. These operators are used to identify specific states in a superposition.

Now given a set $\wp$ of m binary patterns of length n to be memorized, 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 analogously as n neurons in a quantum associative memory. For convenience, the qubits are arranged in three quantum registers labeled x, g, and c, and the quantum state of all three registers together is represented in the Dirac notation as $|x, g, c\rangle$.

The x register will hold a superposition of the patterns. There is one qubit in the register for each bit in the patterns to be stored, and therefore any possible input can be represented. The g register is a garbage register used only in identifying a particular state. It is restored to the state $|\bar{0}\rangle$ after every iteration. The c register contains two control qubits that indicate the status of each state at any given time. A high-level intuitive description of the algorithm is as follows. The system is initially in the single basis state $|\bar{0}\rangle$. The qubits in the x register are selectively flipped so that their states correspond to the inputs of the first pattern. Then, the state in the superposition representing the pattern is "broken" into two "pieces" -- one "larger" and one "smaller" and the status of the smaller one is made permanent. Next, the x register of the larger piece is selectively flipped again to match the input of the second pattern, and the process is repeated for each pattern. When all the patterns have been "broken" off of the large "piece", then all that is left is a collection of small pieces, all the same size, that represent the patterns to be stored; in other words, a coherent superposition of states is created that corresponds to the patterns, where the amplitudes of the states in the superposition are all equal. The algorithm requires $O(mn)$ steps to encode the patterns as a quantum superposition over n quantum neurons. Note that this is optimal in the sense that just reading each instance once cannot be done any faster than $O(mn)$.

5 Quantum Associative Memory

A quantum associative memory (QuAM) can now be constructed from the two algorithms of sections 3 and 4. Define the $\hat{P}$ operator as the operator combination of equations (7-9) that implements the algorithm for memorizing patterns described in section 4. Then the operation of the QuAM can be described as follows. Memorizing a set of patterns is simply

$$|\psi\rangle = \hat{P}|\bar{0}\rangle, \tag{10}$$

with $|\psi\rangle$ being a quantum superposition of basis states, one for each pattern. Now, suppose we know $n-1$ bits of a pattern and wish to recall the entire pattern. Assuming that there are not two patterns that differ only in the last bit, we can use Grover's algorithm to recall the pattern as (τ is the target pattern)

$$|\psi'\rangle = \hat{G}\hat{I}_\tau|\psi\rangle \tag{11}$$

applied recursively $\frac{\pi}{4}\sqrt{N}$ times. Thus, with $2n+1$ neurons (qubits) the QuAM can store up to $N=2^n$ patterns in $O(mn)$ steps and requires $O(\sqrt{N})$ time to recall a pattern.

A very simple example will help clarify. Suppose that we have a set of patterns $\wp$ = {000,011,100,110}. Then using equation (10) memorizes the pattern set as the quantum state

$$\hat{P}|\bar{0}\rangle = \frac{1}{2}|000\rangle + \frac{1}{2}|011\rangle + \frac{1}{2}|100\rangle + \frac{1}{2}|110\rangle. \tag{12}$$

Now suppose that we want to recall the pattern whose first two bits were 10. Applying equation (11) gives

$$\begin{aligned} &\hat{G}\hat{I}_\tau\left(\frac{1}{2}|000\rangle + \frac{1}{2}|011\rangle + \frac{1}{2}|100\rangle + \frac{1}{2}|110\rangle\right) \\ &= \hat{G}\left(\frac{1}{2}|000\rangle + \frac{1}{2}|011\rangle - \frac{1}{2}|100\rangle + \frac{1}{2}|110\rangle\right) \\ &= |100\rangle, \end{aligned} \tag{13}$$

and we have thus achieved our goal. We can now observe the system to see that the completion of the pattern 10 is 100.

Using some concrete numbers, assume that $n = 2^4$ and $m = 2^{14}$ (we let m be less than the maximum possible 2^{16} to allow for some generalization and to avoid the contradictory patterns that would otherwise result). Then the QuAM requires $O(mn) = O(2^{18}) < 10^6$ operations to memorize the patterns and $O(\sqrt{N}) = O(\sqrt{2^{16}}) < 10^3$ operators to recall a pattern. For comparison, in [1] Barenco gives estimates of how many operations might be performed before decoherence for various possible physical implementation technologies for the qubit. These estimates range from as low as 10^3 (electrons in GaAs and electron quantum dots) to as high as 10^{13} (trapped ions), so our estimates fall comfortably into this range, even near the low end of it. Further, the algorithm would require only $2n+1 = 2*16+1 = 33$ qubits! For comparison, a classical Hopfield type network used as an associative memory has a saturation point around $0.15n$. In other words, about $0.15n$ patterns can be stored and recalled with n neurons. Therefore, with n=16 neurons, a Hopfield network can store only $0.15*16 \approx 2$ patterns. Conversely, to store 2^{14} patterns would require that the patterns be close to 110,000 bits long and that the network have that same number of neurons.

Grover's original algorithm only applies to the case where all basis states are represented in the superposition equally to start with and one and only one basis state is to be recovered. In other words, strictly speaking, the original algorithm would only apply to the case when the set $\wp$ of patterns to be memorized included all possible patterns of length n and when we new all n bits of the pattern to be recalled -- not a very useful associative memory. However, several other papers have since generalized Grover's original algorithm and improved on his analysis to include cases where not all possible patterns are represented and where more than one target state is to be found [3] [2] [8]. Strictly speaking it is these more general results which allow us to create a useful QuAM that will associatively recall patterns.

Finally, it is worth mentioning that very recently Chuang et. al. have succeeded in physically implementing Grover's algorithm for the case n=2 using nuclear magnetic resonance technology on a solution of chloroform molecules [4]. It is therefore not unreasonable to assume that a small quantum associative memory may be implemented in the not too distant future.

6 Concluding Comments

A unique view of the pattern completion problem is presented that allows the proposal of a quantum associative memory with exponential storage capacity. It employs simple spin-1/2 quantum systems and represents patterns as quantum operators. This approach introduces a large new field to which quantum computation may be applied to advantage -- that of neural networks. In fact, it is the authors' opinion that this application of quantum computation will, in general, demonstrate much greater returns than its application to more traditional computational tasks (though Shor's algorithm is an obvious exception). We make this conjecture because results in both quantum computation and neural networks are by nature probabilistic and inexact, whereas most traditional computational tasks require precise and deterministic outcomes.

The most urgently appealing future work suggested by the result of this paper is, of course, the physical implementation of the algorithm in a real quantum system. As mentioned in section 5, the fact that very few qubits are required for non-trivial problems together with the recent physical realization of Grover's algorithm helps expedite the realization of quantum computers performing useful computation. In the mean time, a simulation of the quantum associative memory is being developed to run on a classical computer at the cost of an exponential slowdown in the length of the patterns. Thus, association problems that are non-trivial and yet small in size will provide interesting study in simulation. Another obvious and important area for future research is investigating further the application of quantum computational ideas to the field of neural networks -- the discovery of other quantum computational learning algorithms. Further, techniques and ideas that result from developing quantum algorithms may be useful in the development of new classical algorithms. Finally, the process of understanding and developing a theory of quantum computation provides insight and contributes to a furthering of our understanding and development of a general theory of computation.

References

[1] Barenco, Adriano, "Quantum Physics and Computers", *Contemporary Physics*, vol. 37 no. 5, pp. 375-89, 1996.

[2] Biron, David, Ofer Biham, Eli Biham, Markus Grassl and Daniel A. Lidar, "Generalized Grover Search Algorithm for Arbitrary Initial Amplitude Distribution", to appear in the *Proceedings of the 1st NASA International Conference on Quantum Computation and Quantum Communications*, February 1998.

[3] Boyer, Michel, Gilles Brassard, Peter Høyer and Alain Tapp, "Tight Bounds on Quantum Searching", *Workshop on Physics and Computation*, pp. 36-43, November 1996.

[4] Chuang, Isaac, Neil Gershenfeld and Mark Kubinec, "Experimental Implementation of Fast Quantum Searching", *Physical Review Letters*, vol. 80 no. 15, pp. 3408-11, April 13, 1998.

[5] Deutsch, David and Richard Jozsa, "Rapid Solution of Problems by Quantum Computation", *Proceedings of the Royal Society, London A*, vol. 439, pp. 553-8, 1992.

[6] Deutsch, David, "Quantum Theory, The Church-Turing Principle and the Universal Quantum Computer", *Proceedings of the Royal Society*, London A, vol. 400, pp. 97-117, 1985.

[7] Feynman, Richard P., R. B. Leighton and Mark Sands, *The Feynman Lectures on Physics*, vol. 3, Addison-Wesley Publishing Company, Reading Massachusetts, 1965.

[8] Grover, Lov K., "Quantum Search on Structured Problems", to appear in the *Proceedings of the 1st NASA International Conference on Quantum Computation and Quantum Communications*, February 1998.

[9] Grover, L., "A Fast Quantum Mechanical Algorithm for Database Search", *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pp. 212-19, 1996.

[10] Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the National Academy of Scientists*, vol. 79, pp. 2554-2558, 1982.

[11] Kosko, Bart, "Bidirectional Associative Memories", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, pp. 49-60, 1988.

[12] Perus, Mitja, "Neuro-Quantum Parallelism in Brain-Mind and Computers", *Informatica*, vol. 20, pp. 173-83, 1996.

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

[14] Simon, D., "On the Power of Quantum Computation", *SIAM Journal of Computation*, vol. 26 no. 5, pp. 1474-83, 1997.

[15] Ventura, Dan and Tony Martinez, "Quantum Associative Memory with Exponential Capacity", *Proceedings of the International Joint Conference on Neural Networks*, pp. 509-13, May 1998.

[16] Ventura, Dan and Tony Martinez, "Initializing a Quantum State's Amplitude Distribution", submitted to *Physical Review Letters*, June 18, 1998.