RESEARCH ARTICLE

The Quantum Binding Problem in the Context of Associative Memory

Andreas Wichert*

Department of Computer Science and Engineering, INESC-ID & Instituto Superior Técnico, Universidade de Lisboa, Porto Salvo, Portugal

* andreas.wichert@tecnico.ulisboa.pt

Abstract

We present a method to solve the binding problem by using a quantum algorithm for the retrieval of associations from associative memory during visual scene analysis. The problem is solved by mapping the information representing different objects into superposition by using entanglement and Grover's amplification algorithm.

Citation: Wichert A (2016) The Quantum Binding Problem in the Context of Associative Memory. PLoS ONE 11(9): e0162312. doi:10.1371/journal.pone.0162312

Editor: Andrew Adamatzky, University of the West of England, UNITED KINGDOM

Received: March 18, 2016

Accepted: August 19, 2016

Published: September 7, 2016

Data Availability Statement: All relevant data are within the paper.

Funding: This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The author has declared that no competing interests exist.

Introduction

Quantum machine learning by a quantum computer can take logarithmic time in the number of vectors and their dimension [1–4]. This time is a significant exponential speed-up over classical algorithms; however, such a speed-up requires quantum input and quantum output [3]. In the absence of quantum input, the data must be read, which results in a linear time complexity. We can use Grover's algorithm to achieve a quadratic speed-up without requiring the output states to be quantum states. However, most quantum machine learning algorithms and quantum associative memories that are based on Grover's algorithm suffer from the unsolved reading destruction problem (RD problem) [3–5]:

- The reading problem: The amplitude distribution of a quantum state is initialized by reading N data points. Although the existing quantum algorithm requires only $O(\sqrt{N})$ steps and is faster than the classical algorithms, N data points must be read. Hence, the complexity of the algorithm does not improve and is $O(N) = O(N) + O(\sqrt{N})$.
- The destruction problem: A quantum associative memory [6–8] for N data points of dimension m requires only m or fewer units (quantum bits). However, this memory can be queried only once because of the collapse during measurement (destruction); hence, quantum associative memory does not have any advantages over classical memory.

We identify a quantum algorithm with a Wilshaw's associative memory model [9, 10] that does not suffer from the RD problem. The visual (sub-symbolic) variable binding algorithm for Wilshaw's associative memory involves intensive computation because it corresponds to a combinatorial problem [11, 12]. We suggest some mechanisms based on quantum computation to reduce this high complexity and present a hybrid architecture from the quantum computation perspective. In the proposed architecture, different objects are mapped into superposition representing different combinations. Wilshaw's associative memory performs the familiarity discrimination task [13] that can be efficiently represented by a quantum oracle. Grover's algorithm considerably improves the time complexity. The main contributions of this interdisciplinary paper are the relation of sub-symbolic variable binding, the usage of Wilshaw's associative memory for familiarity discrimination and its integration into a hybrid quantum computation architecture.

This paper comprises a classical section that reviews previously published work [11, 12, 14] related to the sub-symbolic variable binding approach and an integration section that describes the new quantum computational approach.

Materials and Methods

Sub-symbolic Binding

The binding problem determines a method to connect all of the physically separated fragments of a complex object to enable them to be processed as a whole by an agent. For example, a red block is obviously a different object from a blue block. The binding problem can be divided into two subproblems: the segregation problem and the combination problem. The segregation problem involves the determination of a method to segregate the elements in an input such that they represent objects that can also be features. The combination problem involves the determination of the elements that represent an object or a category. In sub-symbolic binding, the elements are represented by sub-symbols. The sub-symbolic representation often corresponds to a pattern that mirrors the manner in which the biological sense organs describe the world. Patterns are represented by vectors. Thus, the vectors correspond to sub-symbols. An example of a sub-symbolic binding problem is the definition of a category of objects in a visual scene [11, 12].

Sub-symbolic combination problem. Suppose 7 objects were recognized in the visual scene. We represent the 7 objects at various positions in the scene by the symbols *A*, *B*, *C*, *D*, *E*, *F*, *G*. The task is the identification of a category that is formed by the objects represented by the set *B*, *C*, *G* [11, 12, 14]. We determine whether each of the symbols *B*, *C*, *G* is present in the set that represents the scene. We also verify whether a set representing a category is a subset of the set representing a scene. This task is described by proto logic sets [14]. Proto logic operates on sets; it verifies whether a subset is present in a certain set. The proto logic task seems trivial in the case of sets and symbols. However, if the category (set of sub-symbols) is stored in an associative memory, the task is non-trivial and is an example of the combination problem. The combination problem determines a method to combine individual objects or features into a single category. In an associative memory, we do not have direct access to the stored information.

An associative memory operates on vectors of fixed dimensions. Two of these vectors are always associated; this process of association is called learning. The first vector is called the address vector, and the second vector is called the retrieved vector. After the learning process, the address vector is presented to the associative memory, and the retrieved vector is determined. This process is called association. A distinction exists between heteroassociation and auto-association. An auto-association is present if the retrieved vector represents the reconstruction of the faulty address vector. A heteroassociation is present if the retrieved vector is different from the address vector. In our model, we store auto-associations, i.e., the address vector is the same as the retrieved vector. After a retrieved vector is determined, the similarity between the determined retrieved vector and the address vector is calculated. A greater similarity indicates a higher probability of the corresponding address vector being stored in the associative memory.

A set of objects (a category) is represented by a vector created by concatenating the sub-vectors that represent the objects. For M sub-vectors, the number of possible orderings of the corresponding sub-vectors is $M!$. To verify whether a set of M sub-vectors representing a category is a subset of the set of N sub-vectors representing a scene, there are

$$L = Perm(N, M) = \frac{N!}{(N-M)!} \tag{1}$$

L possible orderings [11, 12, 15]. For $N = 7$ and $M = 3$, we must pose $L = 210$ queries to the associative memory, as shown in Fig 1.

The combination problem leads to an combinatorial explosion for large N and M values, for example for $N = 100$ and $M = 4$ the value $9.41094 \cdot 10^7$. From the L permutations only the familiar patterns represent visual categories that were learned.

Wilshaw's Associative Memory

An example for an associative memory is the formal neural net model that integrates the assembly concept [9, 10], also called Lernmatrix or Wilshaw's associative memory. The biological and mathematical aspects of the Wilshaw's associative memory were studied by Wilshaw and Palm [9, 10, 16]. It was shown that Donald Hebb's hypothesis of cell assemblies as a biological model of internal representation of events and situations in the cerebral cortex corresponds to the formal Wilshaw's associative memory model. The Lernmatrix [17, 18] is composed of a cluster of units which represent a simple model of a real biological neuron. The unit is composed of weights which correspond to the synapses and dendrites in the real neuron. They are described by w_{ij} in Fig 2. T is the threshold of the unit. We call the Lernmatrix simply "associative memory" if no confusion with other models is possible. Two pairs of binary vectors are associated, this process of association is called learning. The first of the two vectors is called the address vector and the second, the retrieved vector. After learning, the address vector is presented to the Lernmatrix and the retrieved vector is determined.

Learning. In the initialization phase of the associative memory, no information is stored. Because the information is represented in weights, they are all initially set to zero. In the learning phase, pairs of binary vector are associated. Let **x** be the address vector and **y** the retrieved vector, the learning rule is:

$$w_{ij}^{new} = \begin{cases} 1 & if \quad y_i \cdot x_j = 1 \\ w_{ij}^{old} & \text{otherwise} \end{cases} \tag{2}$$

Retrieval. In the *one-step* retrieval phase of the associative memory, a fault tolerant answering mechanism recalls the appropriate answer vector for a address vector **x**. For the presented address vector **x**, the most similar learned $\mathbf{x}^l$ address vector regarding the Hamming distance is determined and the appropriate retrieved vector **y** is identified. For the retrieval rule, the knowledge about the correlation of the components is sufficient. The retrieval rule for the determination of the retrieved vector **y** is:

$$y_i = \begin{cases} 1 & \sum_{j=1}^{n} w_{ij} x_j \geq T \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where T is the threshold of the unit. The threshold is set as proposed by [19] to the maximum

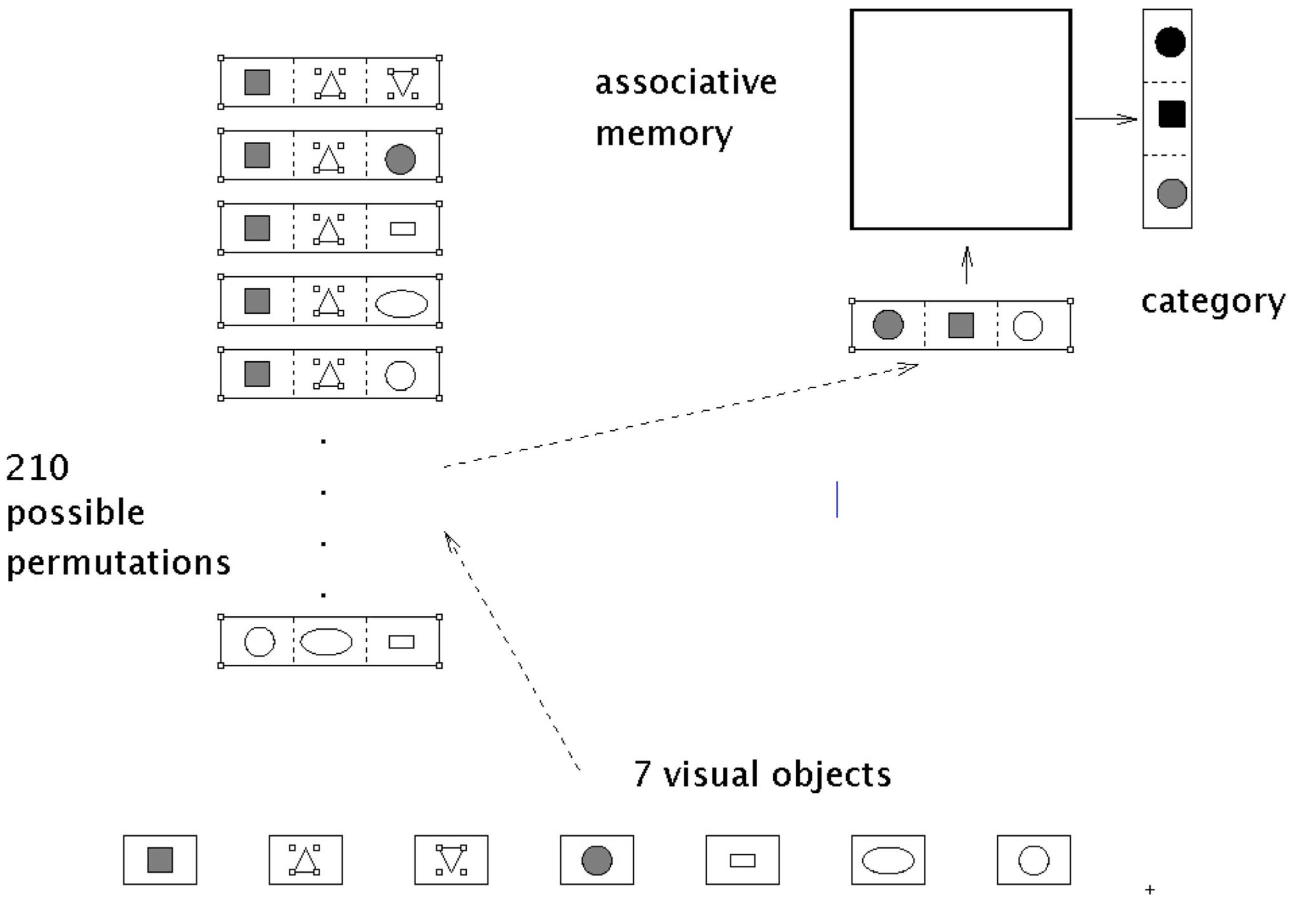


Fig 1. Retrieval phase. In the retrieval phase *L* permutations are formed. Each permutation represents a address vector $\mathbf{x}_i$, $i \in \{1, \ldots, L\}$. Some of the address vectors represent a category, given that the determined retrieved vector is similar to the address vector.

doi:10.1371/journal.pone.0162312.g001

of the sums $\sum_{j=1}^{n} w_{ij}x_j$:

$$T := \max_{1 \le i \le B} \left\{ \sum_{j=1}^{n} w_{ij}x_j \right\}. \tag{4}$$

Only the units which are maximal correlated with the address vector are set to one.

Familiarity Discrimination

For the computation of the reliability of the the answer for the heteroassociation a backward projection of the associative memory is required [12]. The backward projection corresponds to a bidirectional associative memory (BAM) [20]. This time the learned matrix is cued with the retrieved vector and the best address vector is retrieved. Formally, **y** is the address vector, and the retrieved vector which should be determined is $\mathbf{x}^l$. The categorization rule for the determination of the retrieved vector $\mathbf{x}^l$ is:

$$x_j^l = \begin{cases} 1 & \sum_{i=1}^{m} w_{ij}y_i \ge T^* \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

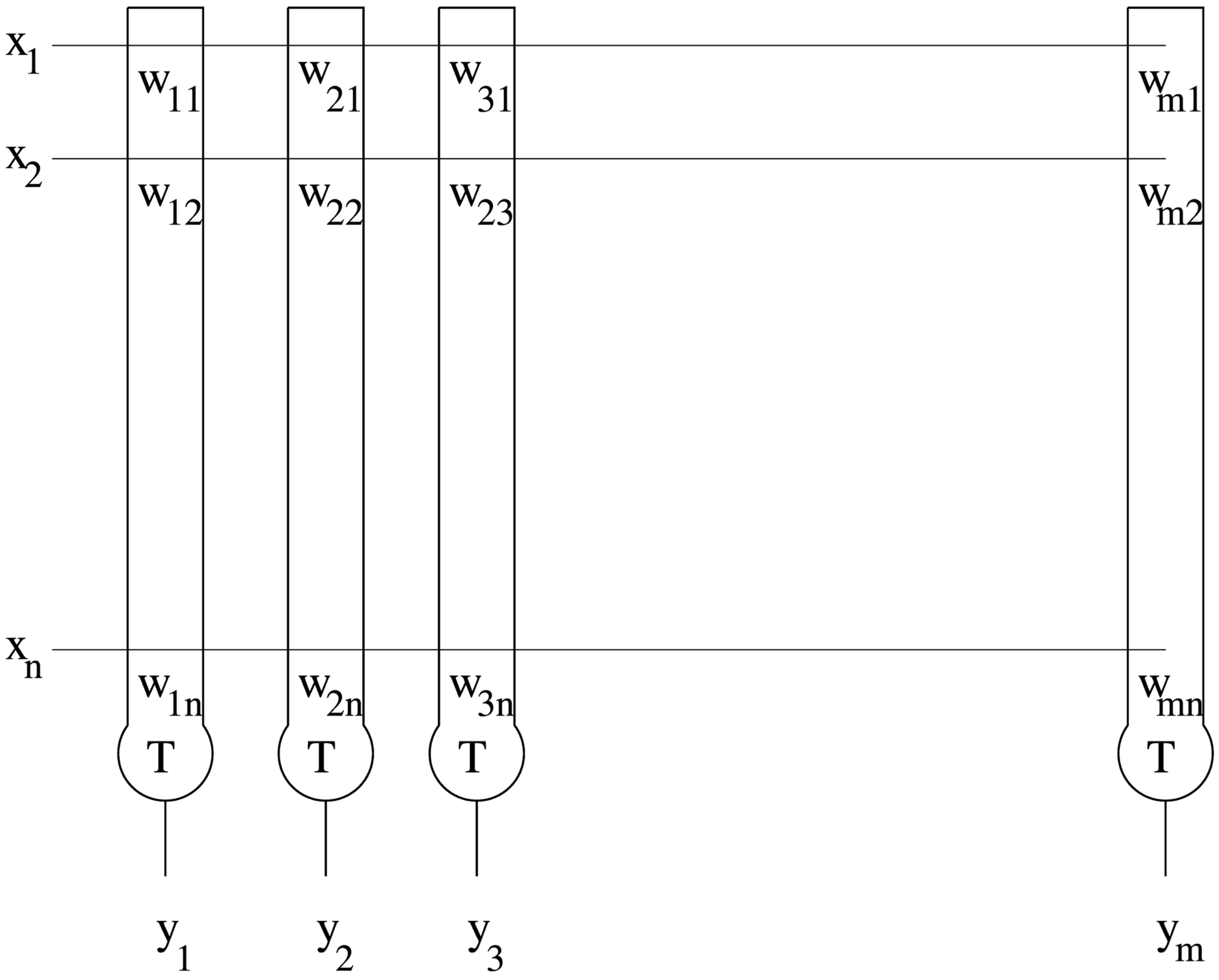


Fig 2. The associative memory. The associative memory is composed of a cluster of units.

doi:10.1371/journal.pone.0162312.g002

This means that the synaptic matrix used is a transposition of the matrix which is used for the forward projection. T^* is the threshold of the unit. The threshold is set to the maximum sum $\sum_{j=1}^{m} w_{ij} y_j$:

$$T^* := \max_{1\leq j\leq A}\left\{\sum_{i=1}^{m} w_{ji} y_i\right\}. \tag{6}$$

Let **x** be the question vector and **y** the retrieved vector that was determined by the associative memory for example by a part of the associative memory. First, the vector $\mathbf{x}^l$ which belongs to the vector **y** is determined. These two vectors form together a vector pair $\mathbf{x}^l$ **y** which is stored in the associative memory. It was either created by learning, $\mathbf{x}^l$ and **y** were learned together, or created through overlap with other already learned vector pairs. The vector $\mathbf{x}^l$ is determined by a backward projection of the vector **y**. In the second step, the similarity of the stored address vector $\mathbf{x}^l$ to the actually presented vector **x** is determined. The greater the similarity of the vector $\mathbf{x}^l$ to the vector **x**, the more reliable the retrieved vector **y**. We can measure the similarity by the Hamming distance function

$$d_1(\mathbf{x},\mathbf{x}^l) = \| \mathbf{x} - \mathbf{x}^l \|_1 = |x_1 - x_1^l| + |x_2 - x_2^l| + \cdots + |x_m - x_m^l| \tag{7}$$

or by the scalar product

$$\langle \mathbf{x} | \mathbf{x}^l \rangle = \cos\omega \cdot \| \mathbf{x} \| \cdot \| \mathbf{x}^l \|, \tag{8}$$

that measure of the projection of one vector onto another.

For auto-association the task there is no need for a backward projection. In the case of auto-association $n = m$. We can measure the similarity by the scalar product with

$$y = W \cdot \mathbf{x} \tag{9}$$

and

$$\langle \mathbf{x} | \mathbf{y} \rangle \tag{10}$$

since W is symmetric with $n = m$. This is equivalent to the quadratic form

$$net = \mathbf{x}^\top \cdot W \cdot \mathbf{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot x_i \cdot x_j. \tag{11}$$

The quadratic form can be as well be interpreted as the energy function [13]

$$H = -\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot x_i \cdot x_j. \tag{12}$$

The threshold operation to determine similarity *sim* is applied to the scalar value *net*,

$$sim = \begin{cases} 1 & net \geq t \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

with the threshold

$$t := T^2 \tag{14}$$

sim = 1 indicates that the question vector **x** was stored in the associative memory, *sim* = 0 that is was not. This operation called the familiarity discrimination, in familiarity discrimination there is no need per se to extract the whole answer pattern [13]. In the following we will only preform auto-association.

Visual Scene Coding

The visual system recognizes objects in an image. It was suggested [21] that the brain includes two mechanisms for visual categorization [22]: one for the representation of the object and the other for the representation of the localization [23]. The first mechanism is called the *what* pathway and is located in the temporal lobe. The second mechanism is called the *where* pathway and is located in the parietal lobe. According to this division, the identity of a visual object can be coded apart from its location. A visual scene can be either represented by an image or by objects and their position in the visual field. Objects are represented by pictograms together with their corresponding position in the image. This is a simple form of structured and compressed representation of a mental image.

The definition of a visual category (see Fig 3) is motivated by the verbal category definition that is a set of prototypical features [24], such as red, round and sweet [25, 26].

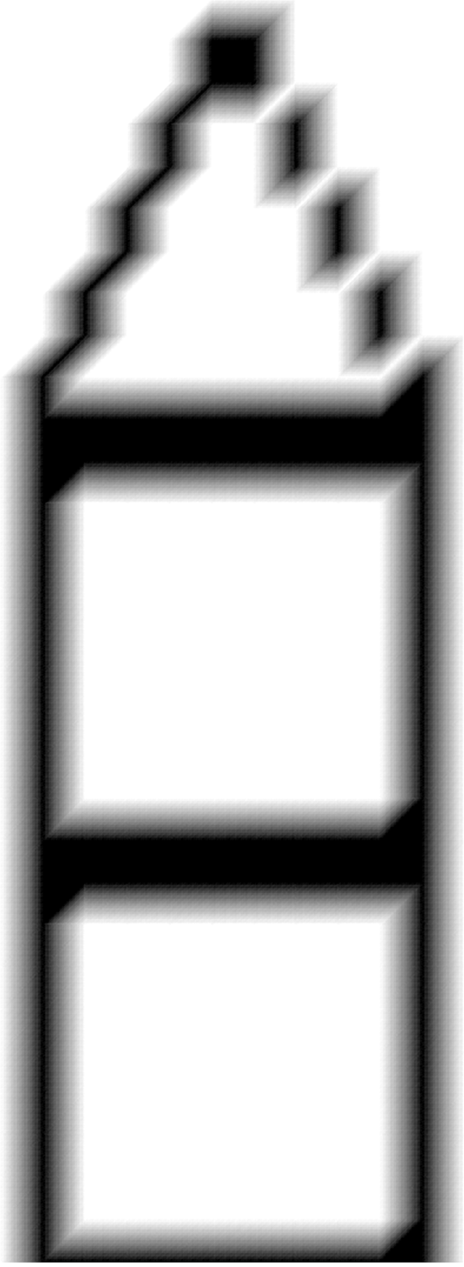


Fig 3. Category. Category "tower" in the blockworld. Blocks can be placed in different positions and picked up and set down. There are two different classes of blocks: cubes and pyramids.

doi:10.1371/journal.pone.0162312.g003

According to [11, 12, 14, 27] the identity of an object can be represented by a binary pattern which is normalized for size and orientation. Its location in the x-axis is represented by a binary vector of the size of the abscissa of the pictogram representing the object. The location in the y-axis is likewise represented by a binary vector of the size of the coordinate of the pictogram representing the object. A binary bar of the size and position of the object in the pictogram of the state represents the location and size (see Fig 4) in each of those vectors. The three vectors that compose the cognitive entity are called associative fields. Each associative field is represented

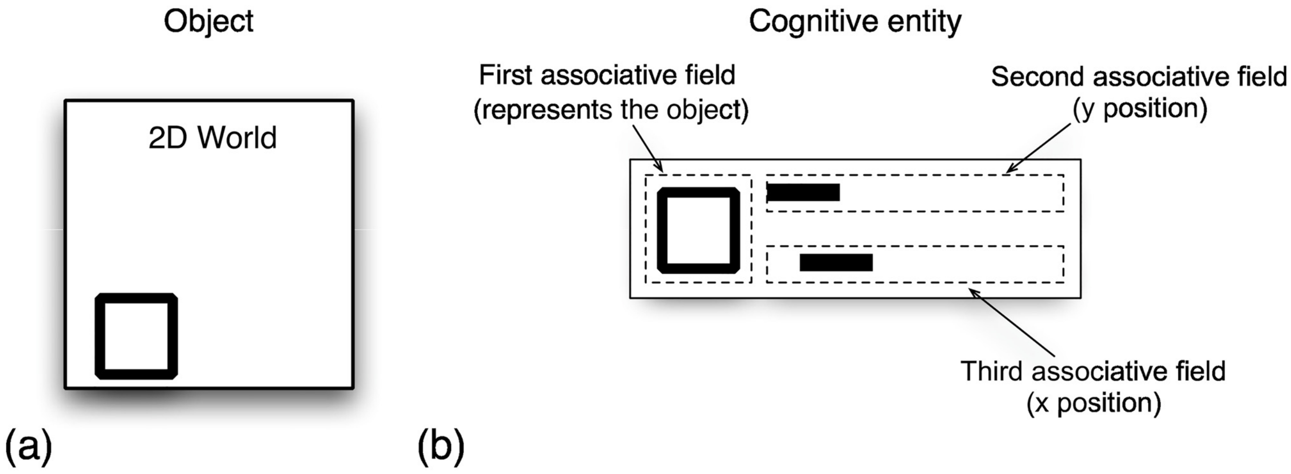


Fig 4. Representation of an object in a 2D world. (a) 2D world. (b). The identity of an object is represented in the first associative field by a binary pattern which is normalized for size and orientation. Its location corresponding to the abscissa is represented by a binary vector in the second associative field. The location corresponding to the ordinate is likewise represented by a binary vector in the third associative field of the size of the ordinate of the pictogram representing the state. A binary bar of the size and position of the object in the pictogram of the state represents the location.

doi:10.1371/journal.pone.0162312.g004

by a binary vector of a fixed dimension; each cognitive entity is formed by the concatenation of the associative fields [11, 12, 14]. A cognitive entity is represented by a binary vector formed by the concatenation of binary vectors which represent the three associative fields [11, 12, 14, 27].

Associations

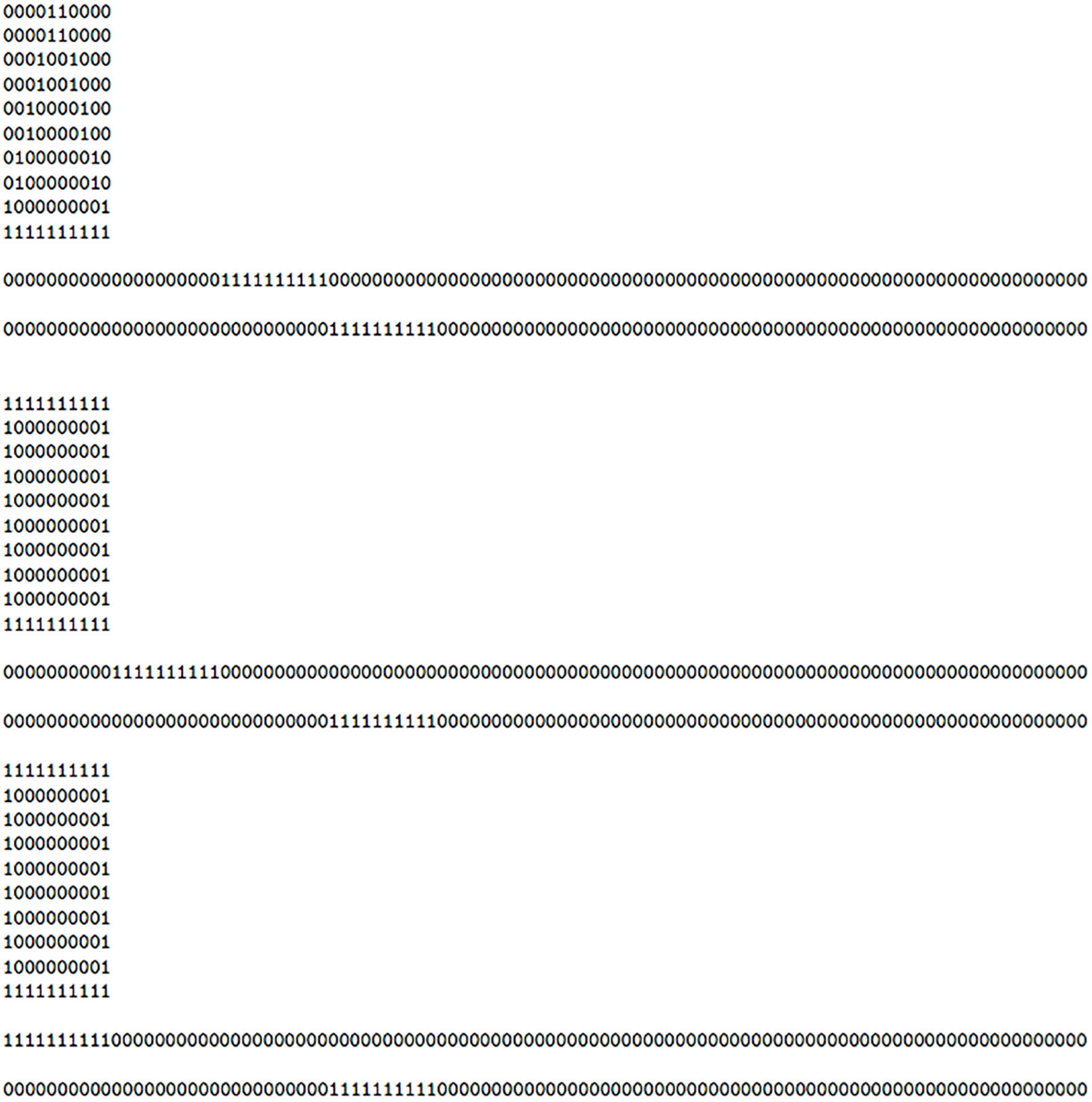
A cognitive entity is represented by a binary vector formed by the concatenation of binary vectors which represent the three associative fields. A visual category "tower" is represented in the blockworld as shown in Fig 3. It corresponds to a set of prototypical visual objects at certain position. The address and retrieved vectors are represented by a binary vector formed by the concatenation of three binary sub-vectors which represent the cognitive entities. Both the question and the answer vectors have dimension 900 because each cognitive entity is described by a binary vector of dimension 300 (= p). The representation of the category "tower" is shown in Fig 5. Associations representing different positions of the category "tower" are learned by the associative memory and can be recognized later despite the presence of noise. Ten associations representing ten different positions of the category "tower" are learned by the associative memory. After learning is complete, a weight matrix of dimension $n = 900$ emerges; the weight matrix consists of three parts, each of size $p = 300$ and $n = 3 \cdot p = 3 \cdot 300$.

Retrieval

In the following example, we indicate the recognition of the category tower (Fig 3) from the pictogram in Fig 6 that is represented by nine different objects. To determine the visual category present, a familiar pattern is determined. During familiarity discrimination, there is no need to extract the entire answer pattern. The corresponding vector values of

$$720 = \frac{10!}{(10-3)!}$$

combinations are determined, and the category tower is identified from the noisy input by



```
0000110000
0000110000
0001001000
0001001000
0010000100
0010000100
0100000010
0100000010
1000000001
1111111111

0000000000000000000011111111110000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000001111111111000000000000000000000000000000000000000000000000000000000000


1111111111
1000000001
1000000001
1000000001
1000000001
1000000001
1000000001
1000000001
1000000001
1111111111

0000000000111111111100000000000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000001111111111000000000000000000000000000000000000000000000000000000000000

1111111111
1000000001
1000000001
1000000001
1000000001
1000000001
1000000001
1000000001
1000000001
1111111111

1111111111000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000001111111111000000000000000000000000000000000000000000000000000000000000
```

Fig 5. Representation by cognitive entities. The representation of the category "tower" by three cognitive entities.

doi:10.1371/journal.pone.0162312.g005

using Eqs 11 and 13 with threshold $t = 0.87$. After determining the correct input vector from the 720 possible combinations, the associative memory is queried, and the answer vector representing the category tower without noise is determined (see Fig 3).

Results and Discussion

Quantum Hybrid Algorithm for Sub-symbolic Binding

In our proposed hybrid approach, we will use Eqs 11 and 13 to build an oracle and Grover's algorithm to speed up the combination of Eq 1. By doing so, we can overcome the RD problem. After determining the correct input vector from all possible combinations, the classical associative memory is queried, and the answer vector representing the category is obtained.

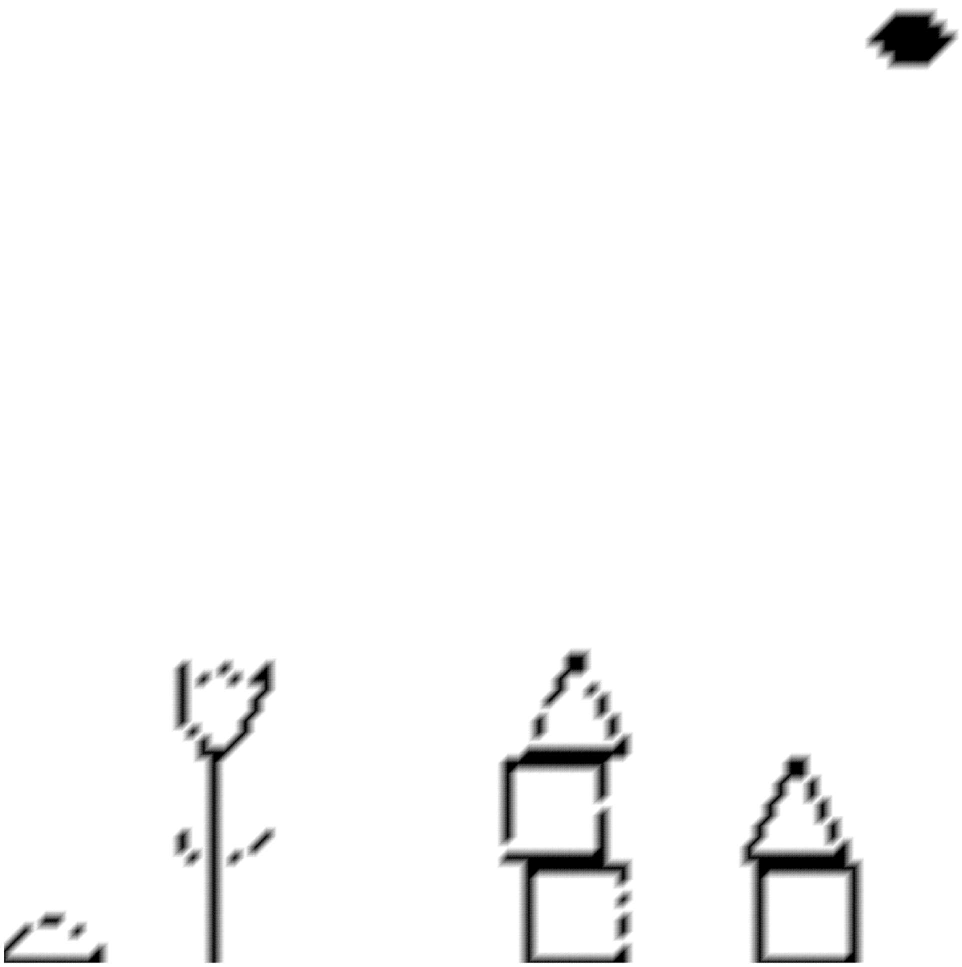


Fig 6. Visual representation by a pictogram. Visual representation of the world with nine different objects and with the category "tower" including noise. The state of the world is described by a pictogram of 100 × 100 pixels; the pictogram is represented by $N = 10$ cognitive entities, and the category tower is described by $M = 3$ cognitive entities.

doi:10.1371/journal.pone.0162312.g006

To simplify the computational process, we can approximate L by permutations with repetition

$$P = N^M \tag{15}$$

where $P > L$. Based on this simplification, we develop a reversible circuit that generates all possible M permutations with repetitions of the N objects. The register $|x\rangle$ should represent the superposition of the N different objects that represent a visual scene; the read time would be $O(N)$. We start with the assumption $M = 1$. In this case, $P = N$, and the superposition corresponds to N objects. Each represented object is identified by a unique address with a value ranging from 1 to N. The task involves loading the corresponding representation of the object having a given address α into the register $|x\rangle$ by a reversible circuit. This task is equivalent to the process of initializing the amplitude distribution of a quantum state, as described in [28, 29]. However, instead of describing the distribution by using a quantum circuit, we load the required values from a given set by using a reversible circuit.

Reversible load. Each object is represented by a vector (of size p) describing the object, the corresponding address and a flag that is set to 0,

$$scene = (\mathbf{z}_1, address_1, flag_1), (\mathbf{z}_2, address_2, flag_2), \cdots, (\mathbf{z}_N, address_N, flag_N)$$

$$scene = (\mathbf{z}_1, 1, 0), (\mathbf{z}_2, 2, 0), \cdots, (\mathbf{z}_N, N, 0).$$

For a given address α, the corresponding address is determined, and the flag is set to one. This task can be performed by a reversible circuit by subtracting the given address α from the unique address, verifying whether the difference is zero and checking whether the result is zero. After this operation, the flag is equal to one. The loading operation is performed by the reversible circuit $load(\alpha, \mathbf{scene})$

$$\mathbf{x}_{address} = load(\alpha, \mathbf{scene}) =: (\mathbf{z}_1 \wedge flag_1) \vee (\mathbf{z}_2 \wedge flag_2) \vee \cdots \vee (\mathbf{z}_N \wedge flag_N).$$

using $N \cdot (N–1) \cdot p$ reversible gates. The reversible circuit corresponds to the reversible operator U_{load}

$$|\mathbf{x}_{address}\rangle|\alpha\rangle|\mathbf{scene}\rangle = U_{load}|0\cdots 0\rangle|\alpha\rangle|\mathbf{scene}\rangle \tag{16}$$

that loads the object representation from $scene$ to a given address α and the resulting waste bits into $|\mathbf{x}_{address}\rangle$. Each binary $\wedge$ and $\vee$ generates one waste bit due to the reversible implementation by a Toffoli gate.

Superposition. The superposition of the address is generated by a Hadamard gate with

$$v = 2^{\lceil log_2 N \rceil} \geq N \tag{17}$$

and

$$|\alpha^*\rangle := H_v \underbrace{|00\cdots 0\rangle}_{v \text{ bits}} = \underbrace{\frac{1}{\sqrt{2^v}}}_{\text{amplitude}} \sum_{x=0}^{2^v-1} \underbrace{|x\rangle}_{v \text{ bits}}. \tag{18}$$

In the case $v > N$ the scene is represented by $v–N$ additional waste objects.

Parallel load. We can perform a parallel loading of the corresponding representation of the object with a certain address represented in the superposition of addresses $|\alpha^*\rangle$ along with the resulting waste bits into the register $|\psi_{\text{x}}\rangle$ by using the reversible operator U_{load}.

$$U_{load}|0\cdots 0\rangle|\alpha^*\rangle|\mathbf{scene}\rangle = U_{load}|0\cdots 0\rangle \otimes H_v|0\cdots 0\rangle|\mathbf{scene}\rangle \tag{19}$$

$$|\psi_{\mathbf{x}}\rangle|\alpha^*\rangle|\mathbf{scene}\rangle = U_{load}|0\cdots 0\rangle|\alpha^*\rangle|\mathbf{scene}\rangle \tag{20}$$

The register $|\psi_{\text{x}}\rangle$ is in superposition because $|\alpha^*\rangle$ is in superposition. Further, it is entangled with the register $|\alpha^*\rangle$ that acts as a pointer to the values represented in $|\mathbf{scene}\rangle$. Because $|\mathbf{scene}\rangle$ is equal in each superposition, it is not entangled with $|\psi_{\text{x}}\rangle$ and $|\alpha^*\rangle$.

Tensor operation. For $M > 1$ we simply tensor $|\psi_{\text{x}}\rangle|\alpha^*\rangle$ M times

$$\underbrace{|\psi_{\mathbf{x}}\rangle|\alpha^*\rangle \otimes |\psi_{\mathbf{x}}\rangle|\alpha^*\rangle \otimes \cdots \otimes |\psi_{\mathbf{x}}\rangle|\alpha^*\rangle}_{M \text{ times}} |\mathbf{scene}\rangle \tag{21}$$

For simplicity we define

$$|\mathbf{x}^*\rangle := \underbrace{|\psi_{\mathbf{x}}\rangle|\alpha^*\rangle \otimes |\psi_{\mathbf{x}}\rangle|\alpha^*\rangle \otimes \cdots \otimes |\psi_{\mathbf{x}}\rangle|\alpha^*\rangle}_{M \text{ times}} \tag{22}$$

it follows

$$|\mathbf{x}^*\rangle|\mathbf{scene}\rangle = \underbrace{|\psi_{\mathbf{x}}\rangle|\alpha^*\rangle \otimes |\psi_{\mathbf{x}}\rangle|\alpha^*\rangle \otimes \cdots \otimes |\psi_{\mathbf{x}}\rangle|\alpha^*\rangle}_{M \quad \text{times}} |\mathbf{scene}\rangle. \tag{23}$$

The register $|\mathbf{x}^*\rangle$ represent P' superpositions of all possible objects M-permutations with repetition of the N objects as well as the entangled addresses and waste bits with

$$P' = v^M = 2^{\lceil log_2 N \rceil^M} \geq N^M. \tag{24}$$

The complexity of the operation is of size $O(N \cdot M)$.

Quantum Oracle for Familiarity Discrimination

The quadratic form can be simplified because the input vector $\mathbf{x}$ and the weight matrix W are binary

$$net = \sum_{i=1}^{n} \sum_{j=i}^{n} w_{ij} \wedge x_i \wedge x_j. \tag{25}$$

This operation requires $2 \cdot n^2$ AND operations and n^2 full adder gates. A full adder is usually a component in a cascade of adders that adds together several bits, for example a four byte adder. A full adder adds two bits together with a bit carried in form another full adder. Its output is one bit together with the bit that is carried out to another full adder (overflow bit).

Reversibles circuit for familiarity discrimination. A reversible full adder can be build out of Toffoli gates. It itself can be represented as the Peres full adder gate [30]. The Peres full adder gate does not change the first input bit x_1. The operation is described by the following mapping on three input bits x_1, x_2, x_3 with $\mathbf{B} = \{0, 1\}$

$$P : \mathbf{B}^4 \rightarrow \mathbf{B}^4 : P(x_1, x_2, x_3, x_4) = (y_1, y_2, y_3, y_4)$$

$$P(x_1, x_2, x_3, x_4) = (x_1, (x_1 \oplus x_2), (x_1 \oplus x_2 \oplus x_3), (x_1 \oplus x_2) \wedge x_3 \oplus x_1 \wedge x_2 \oplus x_4)$$

It computes the full adder operation with the ancilla (fixed) bit x_4 set to 0

$$P(x_1, x_2, x_3, 0) = (x_1, (x_1 \oplus x_2), \underbrace{(x_1 \oplus x_2 \oplus x_3)}_{\text{sum}}, \underbrace{((x_1 \oplus x_2) \wedge x_3 \oplus x_1 \wedge x_2)}_{\text{carry}})$$

bit number x_1 and x_2 are the operands and x_3 is the bit carried in from the previous less significant stage.

To determine if the question vector $\mathbf{x}$ was stored in the associative memory we subtract from the threshold net from t

$$sim = \begin{cases} 1 & (t - net) < \ 0 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

with the threshold

$$t := T^2. \tag{27}$$

A full subtractor can be designed using the same approach as that for an adder. Three bits are involved in performing the subtraction for each bit of the difference: the minuend x_1, the subtrahend x_2, and the borrow in from the previous (less significant) bit order position x_3. The full subtractor gate does not change the first input bit x_1. It performs the full subtractor operation with the ancilla (fixed) bit x_4 set to 0.

$$S(x_1, x_2, x_3, 0) = (x_1, (x_1 \oplus x_2), \underbrace{(x_1 \oplus x_2 \oplus x_3)}_{\text{subtraction}}, \underbrace{(\neg x_1 \wedge (x_2 \vee x_3) \vee x_2 \wedge x_3)}_{\text{borrowed bit}})$$

where bit numbers x_1 and x_2 are the operands, and x_3 is the borrowed bit. The value of *sim* can be determined from the borrow bit in the result; if the borrow bit is equal to one, then *sim* = 1, otherwise *sim* = 0. This operation requires n^2 full subtractor gates. The reversible familiarity discrimination is feasible because it requires $4 \cdot n^2$ reversible gates ($2 \cdot n^2$ and gates, n^2 full subtractor gates and n^2 full adder gates).

The reversible circuit corresponds to the reversible operator U_{Sim}. Permutation is a reversible operation; therefore, $|\mathbf{x}^*\rangle$ can be rearranged in such a way that the operator U_{Sim} accesses only the object representation. In addition to the object description, U_{Sim} requires the weight matrix representation as a binary vector **w**. The results or the sums, ∧ operations, subtraction operations and waste bits are mapped into the register $|\mathbf{result}\rangle$.

$$|\mathbf{result}\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle = U_{Sim} \cdot |0\cdots0\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle \tag{28}$$

A single bit, the borrow bit, determines the familiarity discrimination and whether the subtraction is negative; if the borrow bit is equal to one, then *sim* = 1, otherwise *sim* = 0. We use a circuit U_{CNOT} composed of a controlled not gates to copy the result represented in the borrow bit into a single bit *sim*.

$$|sim\rangle|\mathbf{result}\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle = U_{CNOT} \cdot |0\rangle|\mathbf{result}\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle \tag{29}$$

In quantum computation it is not possible to reset value of the register $|\mathbf{result}\rangle$. Instead we un-compute the former operations of U_{Sim} by applying the U_{Sim}^{-1} operator.

$$|sim\rangle|0\cdots0\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle = U_{Sim}^{-1} \cdot |sim\rangle|\mathbf{result}\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle \tag{30}$$

After these operations, the result of the familiarity operation is represented in the bit $|sim\rangle$, which is in the basis state $|1\rangle$ or $|0\rangle$. The registers $|sim\rangle$ and $|\mathbf{x}^*\rangle$ are entangled. The remaining registers $|\mathbf{scene}\rangle$, $|\mathbf{W}\rangle$ and $|0\cdot\cdot0\rangle$ are not entangled because they are equal. We simplify the notation by defining the global operator U_F. The unitary operator U_F represents our quantum oracle that marks the solution by the bit *sim*; it is composed of the operators U_{Sim}, U_{CNOT} and U_{Sim}^{-1}. Its operation corresponds to

$$|sim\rangle|0\cdots0\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle = U_F \cdot |0\rangle|0\cdots0\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle. \tag{31}$$

Grover's iteration

The number of solutions (in our case, categories) can be determined efficiently by Quantum Fourier Transform [31], [32].

One category present. If one solution is present, Grover's amplification [33], [34], [35], [36], [33], [33] algorithm requires

$$O\left(\left(2^{\lceil log_2 N\rceil}\right)^{\frac{M}{2}}\right) \approx O\left(N^{\frac{M}{2}}\right) = O\left(\sqrt{N^M}\right) \tag{32}$$

applications of the quantum oracle [32], [31] U_F

$$|sim\rangle|0\cdots0\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle = U_F \cdot |0\rangle|0\cdots0\rangle|\mathbf{x}^*\rangle|\mathbf{scene}\rangle|\mathbf{w}\rangle. \tag{33}$$

to determine the correct input vector representing a category that was stored in the associative memory with high probability. After determining the correct input vector, the classic Wilshaw's associative memory is queried, and the vector representing the category without noise is determined.

Several categories present. For r possible categories, only one category is determined with the cost.

$$O\left(\left(\frac{2^{\lceil log_2 N\rceil}}{r}\right)^{\frac{M}{2}}\right) \tag{34}$$

To determine all of the categories, the algorithm can be repeated several times until r different input vectors representing the categories are measured. An alternative approach would be to unlearn the associative memory specified by W from the already recognized category [37].

Cost Analysis

The reversible load requires $N \cdot (N–1) \cdot p$ reversible gates and the reversible familiarity discrimination requires $4 \cdot n^2$ reversible gates with

$$4 \cdot n^2 = 4 \cdot (p \cdot M)^2.$$

Both operation are feasible and require the same number of gates as a classical application. The time complexity of the quantum hybrid algorithm for sub-symbolic binding is significant better then the classical (naïve) approach. For simplification of the computational process L is approximated by P permutations with repetition for one present category. The costs of the algorithm are

$$O\left(\sqrt{N^M}\right) = O\left(\sqrt{N^M}\right) + O(N \cdot M) \tag{35}$$

with

$$\left(2^{\lceil log_2 N\rceil}\right)^{\frac{M}{2}} << L = Perm(N, M) < N^M = P. \tag{36}$$

Conclusion

Using Grover's algorithm, we could achieve a quadratic speed-up without requiring the output states to be quantum states. For example, let us consider a range query vector $\mathbf{y}$ from a collection of N vectors,

$$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_N$$

all vectors $\mathbf{x}_i$ that are ϵ-similar according to the distance function d are searched

$$d(\mathbf{x}_i, \mathbf{y}) < \epsilon. \tag{37}$$

Assuming that the collection of vectors is represented in a superposition, e.g., in $|\psi\rangle$, the cost to determine one possible ϵ-similar vector would be $O(\sqrt{N})$ using the quantum oracle described by Eq 37. The assumption that the speed-up is quadratic is not realistic in practice due to the RD problem; we are required to read N data points and can query only once because of the collapse during measurement (destruction). If the collection of vectors is represented in a superposition, e.g., in $|\psi\rangle$, oould we copy $|\psi\rangle$ before the measurement? An operation that would produce a copy of an arbitrary quantum state such as $|\psi\rangle$ is not possible; we cannot copy non-basis states because of the linearity of quantum mechanics. Due to the collapse during measurement (destruction), no advantage over a classical algorithm can be achieved.

Instead of being represented by units in superposition [6–8], the quantum associative memory is described by a quantum oracle for familiarity discrimination. The input represents permutations of visual objects by superposition. This architecture reduces the computational complexity and represents the relation between the associative memory and a quantum algorithm by a familiarity discrimination task. The proposed approach is a hybrid algorithm because after determining the correct input vector, the classic Wilshaw's associative memory is queried, and the vector representing the category without noise is determined.

Some classical improvements to the classical naive approach were suggested in [11, 12] and involved a more complicated architecture that does not scale up. It is an open question whether such a quantum hybrid approach has any relation with the human brain. It should be noted that the associative memory performs a classical operation and that the quantum aspect involves a familiarity oracle that performs a subconscious determination of the correct input.

In this paper, we present a method to map an algorithm with an associative memory to Grover's amplification algorithm. This mapping may serve as an inspiration for other related algorithms and problems.

Author Contributions

Conceptualization: AW.

Formal analysis: AW.

Investigation: AW.

Methodology: AW.

Project administration: AW.

Software: AW.

Supervision: AW.

Validation: AW.

Visualization: AW.

Writing – original draft: AW.

Writing – review & editing: AW.

References

1. Giovann V, Lloyd S, Maccone L. Quantum Random Access Memory. Physical Review Letters. 2008; 100(16):160501. doi: 10.1103/PhysRevLett.100.160501
2. Harrow A, Hassidim A, Lloyd S. Quantum algorithm for solving linear systems of equations. Phys Rev Lett. 2009; 15:150502.
3. Wittek P. Quantum Machine Learning, What Quantum Computing Means to Data Mining. Elsevier Insights. Academic Press; 2014.
4. Aaronson S. Quantum Machine Learning Algorithms: Read the Fine Print. Nature Physics. 2015; 11:291–293. doi: 10.1038/nphys3272
5. Aïmeur E, Brassard B, Gambs S. Quantum speed-up for unsupervised learning. Machine Learning. 2013; 90:261–287. doi: 10.1007/s10994-012-5316-5
6. Ventura D, Martinez T. Quantum associative memory with exponential capacity. In: Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. vol. 1; 1988. p. 509–513.

7. Ventura D, Martinez T. Quantum associative memory. Information Sciences. 2000; 124(1):273–296. doi: 10.1016/S0020-0255(99)00101-2

8. Tay N, Loo C, Perus M. Face Recognition with Quantum Associative Networks Using Overcomplete Gabor Wavelet. Cognitive Computation. 2010; p. 1–6.

9. Willshaw DJ, Buneman OP, Longuet-Higgins HC. Nonholgraphic associative memory. Nature. 1969; 222:960–962. doi: 10.1038/222960a0 PMID: 5789326

10. Palm G. Neural Assemblies, an Alternative Approach to Artificial Intelligence. Springer-Verlag; 1982.

11. Wichert A, Pereira JD, Carreira P. Visual search light model for mental problem solving. Neurocomputing. 2008; 71(13-15):2806–2822. doi: 10.1016/j.neucom.2007.08.019

12. Wichert A. The Role of Attention in the Context of Associative Memory. Cognitive Computation. 2011; 3 (1). doi: 10.1007/s12559-010-9064-1

13. Sacramento J, Wichert A. Binary Willshaw learning yields high synaptic capacity for long-term familiarity memory. Biological Cybernetics. 2012; 106(2):123–133. doi: 10.1007/s00422-012-0488-4 PMID: 22481645

14. Wichert A. Proto logic and neural subsymbolic reasoning. Journal of Logic and Computation. 2013; 23 (3):627–643. doi: 10.1093/logcom/exs010

15. Wichert A. Sub-Symbols and Icons. Cognitive Computation. 2009; 1(4):342–347. doi: 10.1007/s12559-009-9027-6

16. Palm G. Assoziatives Gedächtnis und Gehirntheorie. In: Gehirn und Kognition. Spektrum der Wissenschaft; 1990. p. 164–174.

17. Steinbuch K. Die Lernmatrix. Kybernetik. 1961; 1:36–45. doi: 10.1007/BF00293853

18. Hecht-Nielsen R. Neurocomputing. Addison-Wesley; 1989.

19. Palm G, Schwenker F, Sommer FT, Strey A. Neural associative memories. In: Krikelis A, Weems C, editors. Associative Processing and Processors. IEEE Press; 1997. p. 307–325.

20. Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall; 1992.

21. Gross CG, Mishkin. The neural basis of stimulus equivalence across retinal translation. In: Harnad S, Dorty R, Jaynes J, Goldstein L, Krauthamer, editors. Lateralization in the nervous system. New York: Academic Press; 1977. p. 109–122.

22. Posner MI, Raichle ME. Images of Mind. New York: Scientific American Library; 1994.

23. Kosslyn SM. Image and Brain, The Resolution of the Imagery Debate. The MIT Press; 1994.

24. Osherson DN. New Axioms for the Contrast Model of Similarity. Journal of Mathematical Psychology. 1987; 31:93–103. doi: 10.1016/0022-2496(87)90038-1

25. Tversky A. Feature of Similarity. Psychological Review. 1977; 84:327–352.

26. McClelland JL, Rumelhart DE. Distributed memory and the representation of general and specific memory. Journal of Experimental Psychology: General. 1985; 114:159–188.

27. Wichert A. Pictorial Reasoning with Cell Assemblies. Connection Science. 2001; 13(1). doi: 10.1080/05940090010022693

28. Ventura D, Martinez T. Initializing the amplitude distribution of a quantum state. Foundations of Physics Letters. 1999; 12(6):547–559. doi: 10.1023/A:1021695125245

29. Rosenbaum D, Perkowski M. Extended superposed quantum-state initialization using disjoint prime implicants. Physical Review A. 2009; 79(6):052310. doi: 10.1103/PhysRevA.79.052310

30. Islam MS, Rahman MM, Begum Z, Hafiz MZ. Low Cost Quantum Realization of Reversible Multiplier Circuit. Information Technology Journal. 2009; 8(2):208–213. doi: 10.3923/itj.2009.208.213

31. Brassard G, Hoyer P, Tapp A. Quantum Counting; 1998. Available from: http://www.citebase.org/abstract?id=oai:arXiv.org:quant-ph/9805082.

32. Brassard G, Hoyer P, Mosca M, Tapp A. Quantum Amplitude Amplification and Estimation. eprint arXiv:quant-ph/0005055. 2000;.

33. Grover LK. A fast quantum mechanical algorithm for database search. In: STOC'96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. New York, NY, USA: ACM; 1996. p. 212–219.

34. Grover LK. Quantum Mechanics helps in searching for a needle in a haystack. Physical Review Letters. 1997; 79:325. doi: 10.1103/PhysRevLett.79.325

35. Grover LK. A framework for fast quantum mechanical algorithms. In: STOC'98: Proceedings of the thirtieth annual ACM symposium on Theory of computing. New York, NY, USA: ACM; 1998. p. 53–62.

36. Grover LK. Quantum Computers Can Search Rapidly by Using Almost Any Transformation. Phys Rev Lett. 1998; 80(19):4329–4332. doi: 10.1103/PhysRevLett.80.4329

37. Wichert A. Learning of Associative Prediction by Experience. Neurocomputing. 2002; 48(1-4):741–42. doi: 10.1016/S0925-2312(01)00671-3