

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316769982>

Neural implementation of operations used in quantum cognition

Article in *Progress in Biophysics and Molecular Biology* · May 2017

DOI: 10.1016/j.pbiomolbio.2017.04.007

CITATIONS

9

READS

362

3 authors:



Jerome R. Busemeyer

Indiana University Bloomington

202 PUBLICATIONS 9,030 CITATIONS

[SEE PROFILE](#)



Pegah Fakhari

Indiana University Bloomington

10 PUBLICATIONS 82 CITATIONS

[SEE PROFILE](#)



Peter Kvam

The Ohio State University

22 PUBLICATIONS 80 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Computational evolution [View project](#)

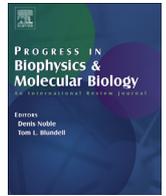


non-commutative processes in psychology [View project](#)



Contents lists available at ScienceDirect

Progress in Biophysics and Molecular Biology

journal homepage: www.elsevier.com/locate/pbiomolbioNeural implementation of operations used in quantum cognition[☆]Jerome R. Busemeyer^{a,*}, Pegah Fakhari^a, Peter Kvam^b^a Indiana University, United States^b Michigan State, United States

ARTICLE INFO

Article history:

Received 24 January 2017

Received in revised form

26 April 2017

Accepted 27 April 2017

Available online xxx

ABSTRACT

Quantum probability theory has been successfully applied outside of physics to account for numerous findings from psychology regarding human judgement and decision making behavior. However, the researchers who have made these applications do not rely on the hypothesis that the brain is some type of quantum computer. This raises the question of how could the brain implement quantum algorithms other than quantum physical operations. This article outlines one way that a neural based system could perform the computations required by applications of quantum probability to human behavior.

© 2017 Elsevier Ltd. All rights reserved.

Contents

1. Confidence judgments during signal detection	00
2. Quantum probability basics	00
2.1. Application to confidence judgments during a signal detection task	00
3. Neural network constraints	00
3.1. Firing rate and membrane potentials	00
3.2. Neural synchrony and quantum inference	00
4. Previous ideas related to neural implementations	00
4.1. Non commutativity of standard neural networks	00
5. A possible neural implementation	00
5.1. Unitary evolution	00
5.2. Choice probability	00
5.3. State reduction	00
6. Future extensions and final comments	00
6.1. Future extensions	00
6.2. Final comments	00
Final comments	00
References	00

Although quantum mechanics is a theory of physics, the mathematics underlying this theory provides the foundation for a general theory of probability (Pitowski, 2006; Suppes, 1966). Most applications of probability theory outside of physics are based on classical theory Kolmogorov (1933/1950). Until recently, quantum probability theory has rarely been applied outside of physics to

fields such as the behavioral and social sciences. However, a body of researchers in the new field called “quantum cognition” have made a reasonably convincing case that quantum probability theory provides a viable new way to formulate theoretical explanations for puzzling behavior that have resisted explanation by classical probability theories (Busemeyer and Bruza, 2012; Khrennikov, 2010). See Ashtiani and Azgomi (2015) for a recent survey of the field.

There are two different views that a quantum cognition researcher can hold regarding the use of quantum probability theory to model human behavior. One view is that quantum

[☆] This research was supported by NSF MMS (SES-1560554 and AFOSR FA9550-15-1-0343).

* Corresponding author.

E-mail address: jbusemey@indiana.edu (J.R. Busemeyer).

probability rules are simply useful for predicting human behavior, and they do not have to be represented at a neurophysiological level (see, e.g., [Atmanspacher and Filk, 2010](#)). The other view is that the brain actually implements these procedures. In particular, the state vector is somehow physically present in the brain.

Researchers who take the view that brain actually implements quantum computations have at least two different ideas about how this can be done. One hypothesis (e.g., [Hammeroff, 1998](#); [Jibu and Yasue, 1995](#))¹ proposes that the brain is using quantum physical mechanisms to represent cognitive states and produce operations. Another hypothesis (e.g., [Eliasmith, 2013](#)) proposes that classical neural network models can implement the computations required by the quantum probability rules. The purpose of this article is to describe a classical neural network that implements quantum computations.

1. Confidence judgments during signal detection

To motivate this presentation, it is helpful to begin with an empirical example that illustrates the kind of evidence used to support the application of quantum probability to human judgment and decision making. One of the key types of findings used to support a quantum interpretation are interference effects, which are essentially violations of the law of total probability.

We recently found evidence for interference effects obtained from a human decision making experiment using a signal detection type task in which a decision maker must decide on each decision trial whether a target is present or absent based on noisy and uncertain information (e.g., to decide whether or not an enemy is located at a position based on a poor and fuzzy image). Decisions are made across several hundred trials – on some trials the signal is present, and on other trials, no signal is present. Accuracy, decision time, and confidence are measured on each trial. Performance on the signal detection task has traditionally been modeled using classical Markov type of random walk/diffusion models of decision-making (see, e.g., [Ratcliff and Smith, 2004](#)). The basic idea is that the decision maker accumulates evidence for each hypothesis until the accumulated evidence reaches a threshold. The first hypothesis to reach the threshold is chosen, the time to reach the threshold determines the decision time, and the difference in evidence soon after the decision determines the confidence ([Pleskac and Busemeyer, 2010](#)).

Alternatively, [Busemeyer et al. \(2006\)](#) developed a quantum walk model for signal detection (summarized later), which assumes that a person's evidence state is represented by a quantum wave function that evolves across levels of confidence in the direction driven by the presented information. [Busemeyer and Bruza \(2012\)](#) derived a key prediction that provides a critical method to empirically distinguish and test the two theories. The experiment consists of two conditions: In the choice-confidence condition, the person makes a choice (makes a binary decision between signal present versus signal absent) at time t_1 and then rates confidence at time t_2 ; in the confidence-alone condition, the person only provides a confidence rating at time t_2 . For both conditions, the focus is on the marginal distribution of confidence ratings that are obtained at time t_2 . Confidence is defined as the judged probability that a signal is present rated on a 0% (certain signal not present) to 100% (certain signal is present) scale. The Markov model obeys the Chapman-Kolmogorov equation, which is a dynamic form of the law of total probability, and it predicts no difference between the two conditions. The quantum model predicts an interference effect

produced by the choice on the confidence rating, which makes the confidence distributions differ between the two conditions.

[Kvam et al. \(2015\)](#) empirically tested for the predicted interference effects by comparing confidence ratings produced by the choice-confidence versus confidence-alone conditions. They obtained strong support for the interference effect predicted by the quantum model. Confidence judgments were, on average, lower in the choice-confidence condition ($M = 83.96$; $SD = 15.56$) than in the confidence-alone condition ($M = 85.15$; $SD = 14.95$), and a Bayesian statistical analysis of the difference resulted in a 95% highest density interval that did not cover zero.² [Fig. 1](#) shows the result for one of the nine participants. The horizontal axis represents the degree of confidence, and the vertical axis represents the relative frequency of reporting a particular level of confidence. Notice the large bump produced by the choice in choice-confidence condition, which is absent for the choice-alone condition. Also notice that the confidence seems to oscillate as it moves up the scale in agreement with the quantum model and contrary to the predictions of the Markov model.

2. Quantum probability basics

Quantum theory was originally developed by a brilliant collection of scientists including Planck, Einstein, de Broglie, Bohr, Heisenberg, Born, Schrödinger and many others, but a firm mathematical foundation was not established until the axiomatic works by Dirac and von Neumann ([Von Neumann, 1932/1955](#); [Dirac, 1930/1958](#)). Of course, the theory has evolved extensively since that time to include new concepts, such as quantum noise decoherence produced by open systems ([Nielsen and Chuang, 2000](#)). However, here we simply describe the very basic ideas. To keep the mathematics at an elementary level, we will restrict our discussion to finite spaces. Although the dimension of the space is finite, it could be very large, e.g., 10 billion, which is less than the number of neurons in the brain! We can translate classical into quantum probability theory as follows.

We start by replacing the classic *sample space* (a finite set of cardinality N) with a quantum *Hilbert space* (a finite vector space of

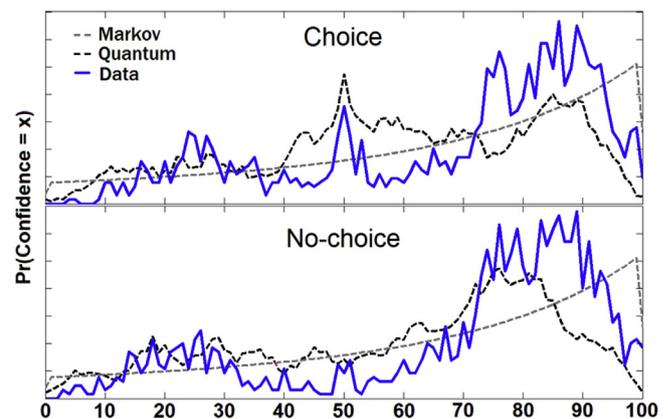


Fig. 1. Interference effects for one participant. Top panel shows choice-confidence condition, bottom panel shows confidence-alone condition. Horizontal axis represents confidence on a 0 = certain absent to 100 = certain present scale. Vertical axis shows relative frequency of a confidence rating. Blue curve shows data, black curve shows quantum predictions, grey curve shows Markov predictions.

¹ The quantum field models of memory by [Freeman and Vitiello \(2006\)](#) seem to lie someplace between an abstract mathematical model and a physical brain model.

² This is the Bayesian version of 95% confidence interval.

dimension N , with an inner product). Next we replace the classic definition of an event – as a *subset* within the sample space – with the quantum concept of event – as a *subspace* within the vector space. Then we replace the classic probability function p with a state vector ψ – the function p maps events into probabilities, and the unit length vector ψ also maps events into probabilities but more indirectly. The quantum probability is computed by the algorithm: $p(A) = \|P(A) \cdot \psi\|^2$, where $P(A)$ is the projector corresponding for the subspace A . Following the measurement of event A , the classical state reduces to the conditional probability function p_A defined by $p_A(B) = \frac{p(A \cap B)}{p(A)}$, and the quantum state reduces to a conditional state $\psi_A = \frac{P(A) \cdot \psi}{\sqrt{p(A)}}$. Finally for dynamics, we replace the classical Kolmogorov forward equation with the Schrödinger equation.

2.1. Application to confidence judgments during a signal detection task

Referring back to the signal detection problem, consider a model for a sequence of confidence judgments during a single trial of this task. For this example, we wish to compute the probability that the cognitive system makes a particular confidence rating, $r_1 \in \{0, 1, \dots, 100\}$, at some point in time t_1 followed by another confidence rating $r_2 \in \{0, 1, \dots, 100\}$ at time t_2 . To model this task, we employ a 101 dimensional Hilbert space. A confidence measurement is represented by 101 unidimensional projectors, $P(r_j)$, for $r_j \in \{0, 1, \dots, 100\}$ which satisfy orthogonality $P(r_i)P(r_j) = 0$ for $i \neq j$ and completeness $\sum P(r_i) = I$. At the start of a decision trial, before the image is presented, the system starts in an uncertain initial state represented by a unit length vector ψ in the 101 dimensional confidence space.

After the image appears, the decision maker's confidence evolves in the direction driven by the evidence for a period of time t_1 . In a quantum model, the Schrödinger equation produces a unitary transformation, $U(t_1)$, that rotates the initial state vector ψ in the direction favoring the evidence for a period of time t_1 to produce a revised state $\psi(t_1) = U(t_1) \cdot \psi$. The projection of the state on the subspace for rating r_1 produces the vector $\phi(t_1) = P(r_1) \cdot \psi(t_1)$, and the probability of this rating equals its squared length $p(r_1) = \|\phi(t_1)\|^2$. The reduced state immediately after observing r_1 equals $\frac{\phi(t_1)}{\sqrt{p(r_1)}}$, which again has unit length.

Then the decision maker continues processing the image, and the state continues to evolve to a new state at time t_2 , $\psi(t_2) = U(t_2 - t_1) \cdot \frac{\phi(t_1)}{\sqrt{p(r_1)}}$. The final confidence rating is evaluated by applying another projector, say $P(r_2)$, to the state $\psi(t_2)$ to produce the projection $\phi(t_2) = P(r_2) \psi(t_2)$ and the probability of choosing this rating equals $p(r_2|r_1) = \|\phi(t_2)\|^2$. Therefore, the sequential probability of choosing rating r_1 and then reporting a rating of say, r_2 , equals

$$\begin{aligned} p(r_1, r_2) &= p(r_1) \cdot p(r_2|r_1) \\ &= \|P(r_2)U(t_2 - t_1)P(r_1)U(t_1) \cdot \psi\|^2. \end{aligned} \quad (1)$$

Longer sequences of measurements can be obtained by extending the products of unitary evolution followed by projection.

One of the key properties generated by the quantum probability rules is what is known as the interference effect, which is a violation of the law of total probability. The classical Markov model for this task obeys the law of total probability:

$$p(r_2) = \sum_{r_1=0}^{100} p(r_1, r_2).$$

But now consider the predictions of the quantum model for the confidence-alone condition at time t_2 .

$$\begin{aligned} p(r_2) &= \|P(r_2) \cdot U(t_2) \cdot \psi\|^2 \\ &= \|P(r_2) \cdot U(t_2 - t_1) \cdot I \cdot U(t_1) \cdot \psi\|^2 \\ &= \|P(r_2) \cdot U(t_2 - t_1) \cdot \sum P(r_j) \cdot U(t_1) \cdot \psi\|^2 \\ &= \sum_{r_1=0}^{100} p(r_1, r_2) + Int \end{aligned} \quad (2)$$

where Int contains the sum of crossproduct terms. If the crossproduct interference terms are zero, then Equation (2) agrees with the law of total probability. However, the interference can be positive, negative, or zero. Human behavioral studies have reported systematic interference effects (violations of total probability) in a variety of different judgment and decision tasks (Pothos and Busemeyer, 2009; Busemeyer et al., 2009; Wang and Busemeyer, 2016). Our goal in this article is to build a plausible neural network model that can implement the evolution of a quantum state, the probability of a response, and the state reduction, as described by the above equations.

3. Neural network constraints

Here we briefly review some of the constraints needed to build a plausible neural network implementation of quantum probability theory.

There are roughly 10 to 20 billion neurons in the neocortex. Each neuron is roughly connected to about 10,000 others. The *connection weight* between two neurons refers to the synaptic efficiency (e.g., the proportion of ion channels opened by a fixed input) for the sending neuron to impact the receiving neuron. Roughly 85% of these connections are *excitatory* (mainly pyramidal neurons), and the remainder *inhibitory* (interneurons). Excitatory connections are always excitatory, and inhibitory connections are always inhibitory; in other words, a neuron can't change from excitatory to inhibitory or visa versa.

The neurons are organized into *micro columns* containing about 100 neurons, and the neurons within these micro columns are highly interconnected with excitatory and inhibitory connections to form a recurrent network. The micro columns themselves are connected together to form *corticocortical* columns containing 100 micro columns. *Macro columns* contain about 100 corticocortical columns.

Each neuron has a *membrane potential* (voltage level). The membrane potential remains at some *resting potential* until driven by inputs. Excitatory inputs increase and inhibitory inputs decrease the membrane potential, and the membrane potential at any time is the net result. The membrane potential must exceed a *threshold* for a brief spike (<1 ms) to occur in a neuron, which is then transmitted to other neurons. After the spike, the neuron returns for a period of time back to (or below) the resting potential. Only about 15% of neurons are active at any time. Spike rate for a micro column of neurons is determined by the balance of excitatory and inhibitory inputs to the membrane potential for a micro column.

Most neural network models rely primarily on spike rate as the carrier of information between neurons. The membrane potential at a micro column is converted into a spike rate, which is sent to other micro columns. Some neural models work directly at the level of individual spiking neurons (e.g., Eliasmith, 2013). For example, to produce a spiking neuron, the net difference between the excitatory and inhibitory potentials are integrated across time until it exceeds a threshold to produce a single spike. Other neural models

work with a firing rate produced by integrating across a micro column (e.g., O'Reilly et al., 2012). The firing rate (e.g., proportion of neurons that have fired within a micro column) is computed as an increasing function of the difference between the excitatory input and a threshold level set by the inhibitory input for a micro column. In either case, eventually, the information coded by spike rate must be decoded and inverted back into a membrane potential at the receiving end.

Neural oscillations form another important property of neural systems (Nunez and Srinivasan, 2006). There is considerable evidence showing that neural synchrony is important for many neural computations (Fries, 2009). For example, theta oscillations (4–8 Hz) increase during both verbal and spatial memory tasks (Kahana et al., 2001), and alpha oscillations (8–13 Hz) are correlated with attention focus (Ward, 2003). Although a single neuron may be incapable of detecting synchronization (Shadlen and Movshon, 1999), synchronous oscillations produced by recurrent neural networks may implement important cognitive processes such as attention, learning, and memory (Grossberg and Versace, 2008).

3.1. Firing rate and membrane potentials

The neural implementation described below uses pairs of real numbers, with positive or negative values, to compute the unitary evolution. The membrane potentials are real valued and with positive or negative values. However, firing rates are strictly non-negative. According to the quantum model, it is important to define the output from unitary evolution in terms of the pairs of real numbers, and define the firing rate by the squared magnitude of the output.

Let us recall the basic processes used in neural communication. The membrane potential for a group of neurons is converted into a firing rate, and only the firing rate is passed along the axons. However, after reaching the target, the firing rate is then decoded back into a membrane potential. Therefore, we postulate that the transformation of the information from firing rate back to membrane potential inverts the original mapping from membrane potential to firing rate, and recovers the original membrane potential that generated the firing rate. Models that describe a mapping from membrane potential to firing rate are usually monotonic and invertible (e.g., O'Reilly et al., 2012). Therefore, we postulate that when working with the output from unitary evolution, we are modeling the membrane potential for a large group of neurons rather than firing rate.

3.2. Neural synchrony and quantum inference

Neural oscillators provide a critical mechanism for implementing the computations required by unitary evolution. The integration of oscillatory activities produced by different neural oscillators converging on a common target provide a way to compute the positive or negative interference patterns expressed in Equation (2). Synchronized oscillators will produce constructive interference, while asynchronous oscillators will produce destructive interference.

Fig. 2 illustrates the basic idea. Imagine two input nodes, ψ_1 and ψ_2 that converge on a common output node ϕ . First, suppose input 1 is a signal $\psi_1(t) = \cos(.15 \cdot 2\pi \cdot t)$, and suppose input 2 is a synchronous signal $\psi_2(t) = \cos(.15 \cdot 2\pi \cdot t)$ at the same frequency with the same phase. In quantum terms, $\psi_1(t)$ is the amplitude corresponding to one path, and $\psi_2(t)$ is the amplitude corresponding to a second path, and both paths sum to produce the total amplitude $\phi(t) = (\psi_1(t) + \psi_2(t)) = 2 \cdot \cos(.15 \cdot 2\pi \cdot t)$ for the final output. The probability of the final output is obtained by squaring

$\phi^2(t) = (\phi_1(t) + \phi_2(t))^2 = 4 \cdot \cos(.15 \cdot 2\pi \cdot t)^2$. The result is shown as the blue curve in the bottom panel on the left hand side of Fig. 2. Notice that the probability of the sum of paths (shown as the blue curve in the bottom left panel), $(\psi_1(t) + \psi_2(t))^2 = 4 \cdot \cos(.15 \cdot 2\pi \cdot t)^2$, is greater than the sum of the probability of each separate path (red curve), $\psi_1^2(t) + \psi_2^2(t) = 2 \cdot \cos(.15 \cdot 2\pi \cdot t)^2$, and so paths produce constructive interference. Now suppose input 1 is the same signal $\psi_1(t) = \cos(.15 \cdot 2\pi \cdot t)$ as before, but input 2 is an asynchronous signal, so that $\psi_2(t) = \cos(.15 \cdot 2\pi \cdot t + 1.8 \cdot \pi/2)$, which is the same frequency but out of phase. The sum of these two signals cancels out as shown on the right hand side of Fig. 2. The probability produced by the sum of these asynchronous paths (shown as the blue curve in the bottom right panel) $\phi^2(t) = (\psi_1(t) + \psi_2(t))^2 = (\cos(.15 \cdot 2\pi \cdot t) + \cos(.15 \cdot 2\pi \cdot t + 1.8 \cdot \pi/2))^2$ is almost zero, and less than the sum of the probabilities of each separate path (red curve), $\psi_1^2(t) + \psi_2^2(t) = \cos(.15 \cdot 2\pi \cdot t)^2 + \cos(.15 \cdot 2\pi \cdot t + 1.8 \cdot \pi/2)^2$, and so the asynchronous signals produce negative interference.

4. Previous ideas related to neural implementations

Does there exist some classical dynamic system that can implement the quantum algorithm? One answer to this question was established by Graben and Atmanspacher (2006), who showed that coarse grained measurements of classical dynamical systems can produce incompatible observables like that used in quantum probability theory.³ However, their theory was based on a general form of classical dynamic systems, and they didn't intend to develop a concrete neural network implementation of the quantum algorithm.

More directly relevant is the work by de Barros and Suppes (2009) and De Barros (2012), who pointed out that neural oscillators (e.g., see Fig. 2) can produce interference patterns like that obtained from unitary evolution. However, they did not formulate a complete neural network model capable of implementing the general unitary evolution algorithm.

Takahashi and Cheon (2012) proposed a nonlinear neural network that used squared amplitudes used to compute the probability of response in a quantum model of cognition. However, they did not propose any specific network for computing the squared magnitudes or for describing the evolution of the quantum state.

Stewart and Eliasmith (2013) claimed that their spiking neural network model can learn a set of connection weights that could closely approximate the input and output behavior produced by a quantum model. However, as they pointed out, their general neural network model is not at all constrained to obey the properties implied by quantum rules— it can also compute results that disobey these rules. We would like to construct a plausible neural network model that is restricted to implement only the quantum rules.

4.1. Non commutativity of standard neural networks

It is simple to show that a standard neural network, containing two or more layers, can produce the kinds of non-commutative processing that is essential for quantum models. Consider the following simple case. Suppose a $n \times 1$ input vector of activation, X , is fed into a feedforward network, represented by a $n \times n$ connection weight matrix, W_1 , to produce a $n \times 1$ net activation $\eta_1 = W_1 \cdot X$, which is nonlinearly transformed, for example, by a logistic type function f to produce a $n \times 1$ output activation $Y_1 = f[\eta_1]$. Then the previous output Y_1 is fed into another network, represented by a $n \times n$ connection weight matrix, W_2 , to produce a

³ This does not imply that it is possible to construct a local and realist classical theory for the Bell type of experiments in physics.

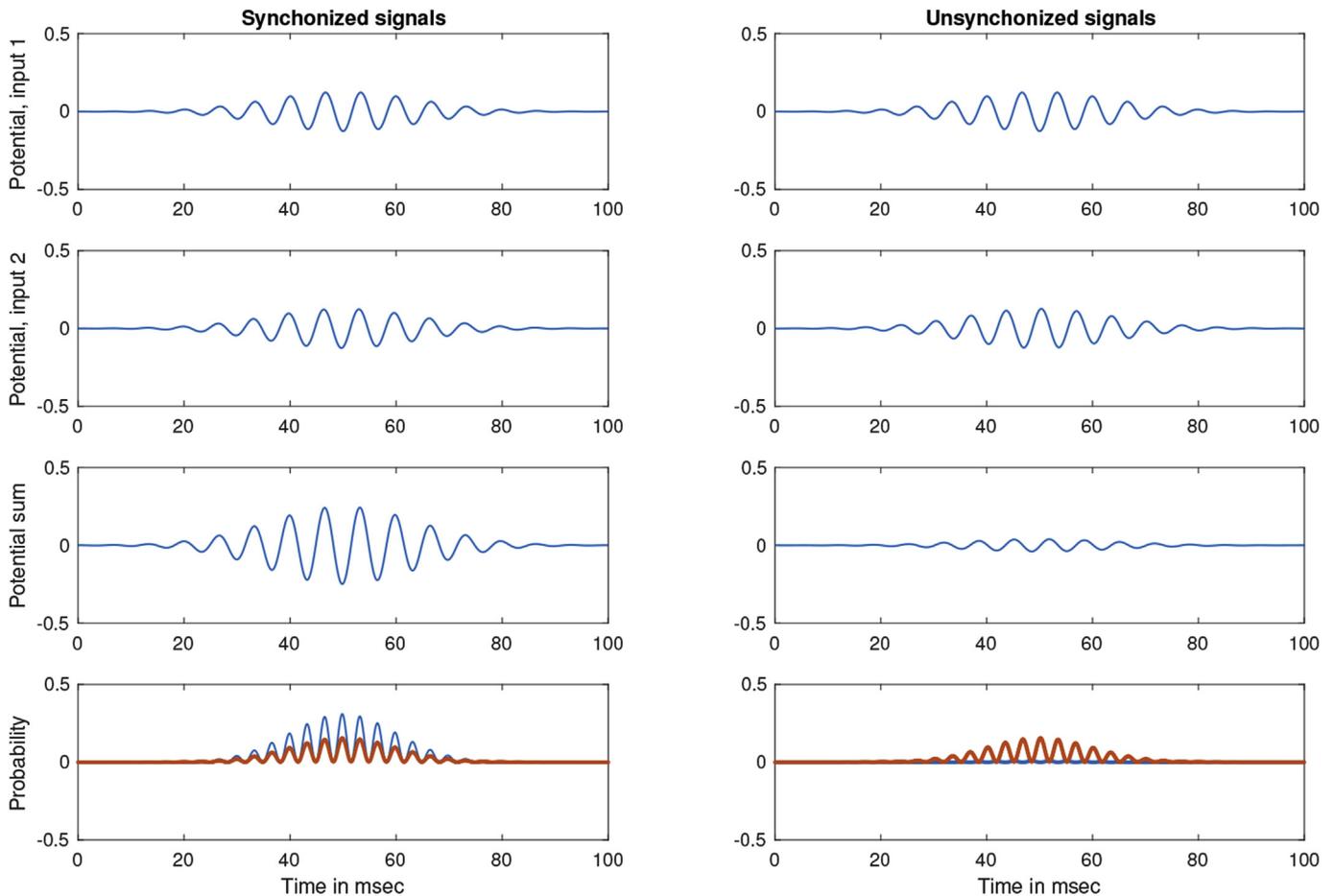


Fig. 2. Effects of synchrony on quantum probability. Left panel shows synchronous signals, and right panel shows asynchronous signals. Top panel shows input 1, second panel shows input 2, third panel shows sum of inputs, fourth panel shows squared sum of inputs (blue) or sum of squared inputs (red). The difference between blue and red in the bottom panel is the interference effect.

net activation $\eta_2 = W_2 \cdot Y_1$, which is nonlinearly transformed by a logistic function to produce a second $n \times 1$ output $Y_2 = f[\eta_2] = f[W_2 \cdot f[W_1 \cdot X]]$. Processing in the opposite order would produce a different result with $Z_2 = f[W_1 \cdot f[W_2 \cdot X]]$.

The fact that neural networks generally produce non-commutativity implies that they are capable of producing all of the hallmark effects used to justify the quantum models, including question order effects, conjunction fallacies, and violations of total probability. Although neural network models can generate non-commutative effects like a quantum model, general networks are unconstrained and they are not required to obey all of the rules that a quantum system must obey. Neural nets can also generate predictions that violate quantum rules.

Specifically, the quantum algorithm is constrained to assign probabilities to sequences of events, where each event is defined by a subspace of the Hilbert space. Furthermore, these probabilities always satisfy the properties of a proper probability measure. In fact, the quantum algorithm is the unique way to do this Gleason (1957) (for Hilbert spaces with dimension greater than 2).

5. A possible neural implementation

The neural network described below is designed to implement the computations required by a quantum cognition model. We develop our neural net using the example of the sequential confidence rating task described earlier.

We now outline the basic steps of one possible neural implementation. This implementation should be considered more like an existence proof rather than an empirically supported neural model. The ideas are admittedly speculative, and there may be many other ways to implement the quantum algorithm. Also this implementation is designed to be more of a macro level rather than micro level representation. Nevertheless, macro level neural network models, similar to this, are frequently used by computational neuroscientists (see., e.g., Verdonck and Tuerlinckx, 2014; Deco et al., 2008).

5.1. Unitary evolution

In order to implement the quantum algorithm as a neural computational model, we need to break down the general equation into more manageable pieces that are easier to implement, and then put the pieces back together in large neural network model. So we start by breaking down the algorithm into simpler-to-compute pieces.

First consider the $N \times N$ unitary matrix $U(t)$. Any unitary matrix can be derived from a Hermitian matrix H by the matrix exponential $U(t) = \exp(-i \cdot t \cdot H)$. The $N \times N$ Hermitian matrix H has a spectral decomposition

$$H = U \cdot \Lambda \cdot U^\dagger = \sum \lambda_j \cdot (U_j U_j^\dagger) = \sum \lambda_j \cdot V(j),$$

where $V(j) = U_j U_j^\dagger$ is the projector for the j -th eigenvector U_j of H , and $j = 1, N$. Therefore, the unitary matrix can be expressed as

$$U(t) = \sum e^{-i \cdot \lambda_j \cdot t} \cdot V(j).$$

The eigenvalues, λ_j are real because H is Hermitian. If, in addition, H is further constrained to be real and symmetric (which has been the case for applications in quantum cognition), then $V(j)$ is a real valued $N \times N$ matrix too. The eigenvalues are called the *phases* of the unitary transformation, and they are important for computing interference effects as shown in Fig. 2.

Using the results from above, the projection can be expressed as

$$\phi(t_1) = P(r_1) \cdot U(t_1) \cdot \psi = \sum_j e^{-i \cdot \lambda_j \cdot t_1} \cdot P(r_1) \cdot V(j) \cdot \psi. \quad (3)$$

How does this help? Well, now we only need to build a neural network to compute for each eigenvector

$$\phi(j) = e^{-i \cdot \lambda_j \cdot t_1} \cdot P(r_1) \cdot V(j) \cdot \psi,$$

and then sum, across eigenvectors, the contributions of these individual neural networks to finally form the output vector $\phi(t_1) = \sum \phi(j)$. Of course, the same type of neural network can then be used to compute the next product $P(r_2) \cdot U(t_2 - t_1) \cdot \frac{\phi(t_1)}{\sqrt{p(r_1)}}$.

Quantum models generally work with complex values, which can be expressed as a pair of real values.⁴ Therefore, we define the input vector ψ as a pair of real valued vectors: $\psi = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, where α is the real part and β is the imaginary part of ψ . Also we can define the complex exponential as cosine/sine pair $e^{-i \cdot \lambda_j \cdot t} = (\cos(\lambda_j \cdot t), -\sin(\lambda_j \cdot t))$. Finally, we define the output vector for the j -th eigenvector as the pair $\phi(j) = \begin{bmatrix} \gamma(j) \\ \omega(j) \end{bmatrix}$, where $\gamma(j)$ is the real part of the output, and $\omega(j)$ is imaginary part. Then using the rules of multiplication for complex numbers, we can express the output shown in Equation (3) as

$$\begin{bmatrix} \gamma(j) \\ \omega(j) \end{bmatrix} = \begin{bmatrix} \cos(\lambda_j \cdot t) \cdot P(r_1) \cdot V(j) \cdot \alpha + \sin(\lambda_j \cdot t) \cdot P(r_1) \cdot V(j) \cdot \beta \\ \cos(\lambda_j \cdot t) \cdot P(r_1) \cdot V(j) \cdot \beta - \sin(\lambda_j \cdot t) \cdot P(r_1) \cdot V(j) \cdot \alpha \end{bmatrix}. \quad (4)$$

Now we only need to construct a neural network for computing $\gamma(j)$, and the same type could be used to compute $\omega(j)$. The network for computing $\gamma(j)$ is itself just the sum of two networks.

The proposed neural network for computing the output vector $\gamma(j)$ is shown in Fig. 3. The network for $\omega(j)$ is similar. This model is a macro level model, and so the circular nodes in the figure represent large groups of interconnected neurons (e.g., micro columns) rather than individual neurons. On the left are shown the pair of inputs to the network. The i -th pair of input nodes for the pair of vectors (α, β) , is shown as the pair of input values (α_i, β_i) . Each pair of inputs (α_i, β_i) is assumed to enter a pair of neural oscillators to produce a pair of oscillations $(\alpha_i \cdot \cos(\lambda_j \cdot t), \beta_i \cdot \sin(\lambda_j \cdot t))$. Each pair of oscillations $(\alpha_i \cdot \cos(\lambda_j \cdot t), \beta_i \cdot \sin(\lambda_j \cdot t))$ is then connected to each of the nodes for the output γ_k of the vector $\gamma(j)$ corresponding to the j -th eigenvector.

$$\gamma_k(j) = \sum_i w_{ki} \cdot \cos(\lambda_j \cdot t) \cdot \alpha_i + w_{ki} \cdot \sin(\lambda_j \cdot t) \cdot \beta_i$$

The connection weight, w_{ki} , connecting input node i from α to output node k shown in the figure represents the matrix element in the i -th column and k -th row of the matrix $P(r_1) \cdot V(j)$.

⁴ The complex number z can be written as $z = x + i \cdot y$, where the real number x is the real part of z and the real number y is the imaginary part. We chose to develop a neural network based on the real x and imaginary y parts. Alternatively, we can write $z = |z| \cdot (\cos(\theta) + i \cdot \sin(\theta))$ and build a neural network on the basis of the magnitude $|z|$ and phase θ . Kak (1995) suggested using the latter approach.

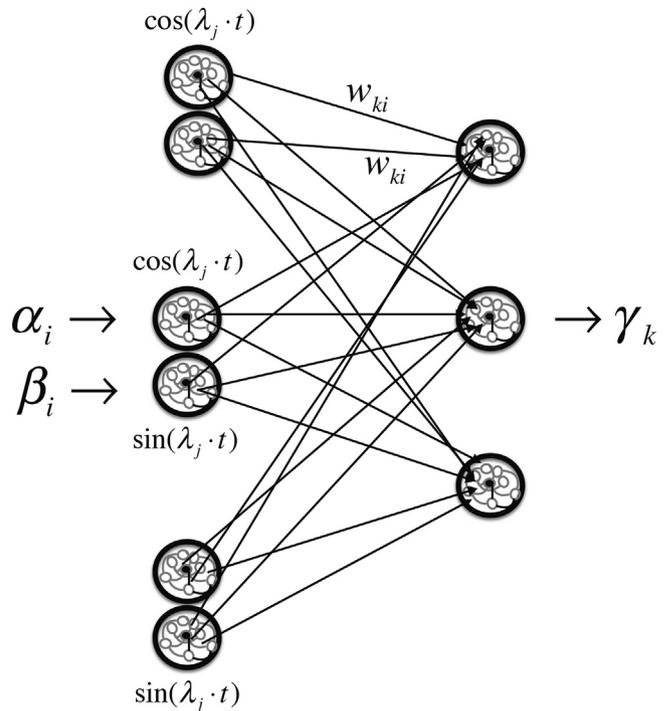


Fig. 3. Neural Network. Left side shows input to network. Middle section shows neural oscillators produced by a recurrent network formed by a group (micro column) of neurons. Right side shows output produced by sum of weighted connections of network.

We can summarize the neural computations as follows. First the type of neural network shown in Fig. 3 is used to compute both $\gamma(j)$ and $\omega(j)$ for each eigenvector U_j using a separate but similar network for each one. A collection of N networks like those shown in Fig. 3 are used to compute the pairs of outputs $(\gamma(j), \omega(j))$ corresponding to each eigenvector U_j of the Hermitian matrix H . The outputs from these N networks are summed across the j eigenvectors to produce the final output $\phi(t) = \begin{bmatrix} \gamma(t) \\ \omega(t) \end{bmatrix}$. This produces the projection, $\phi(t_1) = P(r_1) \cdot U(t_1) \cdot \psi$. The second projection $P(r_2) \cdot U(t_2 - t_1) \cdot \frac{\phi(t_1)}{\sqrt{p(r_1)}}$ is computed using the same type of networks.

5.2. Choice probability

According to the quantum rules, the probability of choosing response r_k at time t_1 equals $p(r_k) = \|P(r_k)\psi(t_1)\|^2 = \|\phi(t_1)\|^2 = |\phi_k|^2$. The next problem to address concerns the question of how a neural system uses the state vector to generate a single choice according to the probabilities given by the quantum rules. First we need to convert each amplitude, ϕ_k , into a firing rate, $|\phi_k|^2 = \gamma_k^2 + \omega_k^2$, for response r_k . Second, we need to provide a neural mechanism that uses the firing rate to choose a single response with a probability given by $p(r_k) = |\phi_k|^2$.

The two coordinates of the amplitude $\phi_k = (\gamma_k, \omega_k)$ are real, but they can be positive or negative. Therefore we need two networks, one for each coordinate, with each network transforming a real value into its squared magnitude. The outputs from the two networks can be summed to produce $|\phi_k|^2$. The top panel of Fig. 4 shows a simple network that takes γ_k (less than one in magnitude) and converts it into an output that closely approximates γ_k^2 .

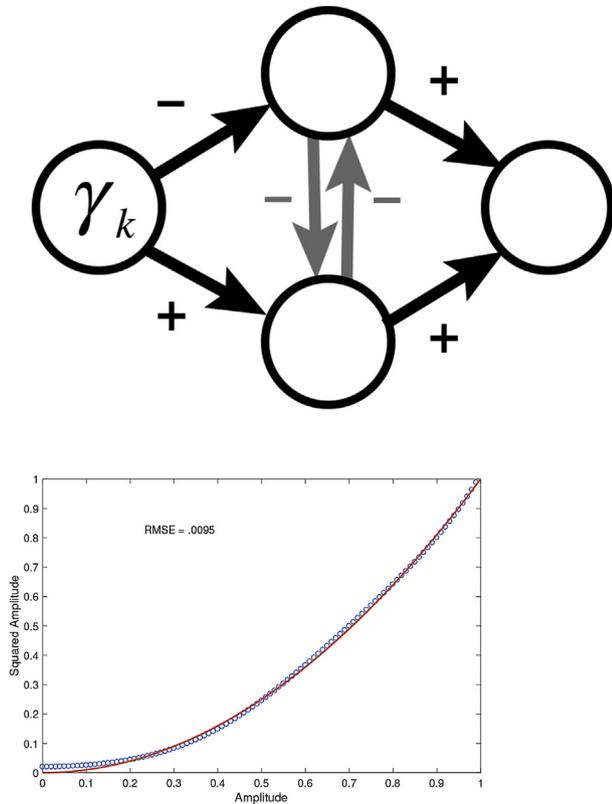


Fig. 4. Left panel: Neural network to convert amplitudes to probabilities; right panel: approximation to squared amplitude produced by network.

(the computer code for this network is shown in the appendix). The input amplitude is passed up through an inhibitory link as well as down through an excitatory link, so that the two middle nodes contain the same magnitude with different signs. Then the recurrent lateral inhibition attenuates the value of the negative middle node and amplifies the value of the positive middle node to generate an approximation to the squared magnitude that is passed to the output on the far right node. The bottom panel of Fig. 4 shows the close approximation to the squared magnitude produced by the network (described in detail in the appendix).

A simple mechanism for choosing a single response on a single trial can be realized by an independent parallel neural activation system. The output $|\phi_k|$ represents the firing rate of a neural node corresponding to response r_k , and the activation produced by this firing rate is accumulated over time. Each response is associated with an independent accumulator, and these accumulators race until one accumulator reaches a threshold for the first time. The first accumulator to reach the threshold wins the race and is chosen. Theorem 2 in Marley and Colonius (1992) states that the probability that the accumulator for response r_k wins the race and

$$p(r_k) = \frac{|\phi_k|^2}{\sum_j |\phi_j|^2}$$

is chosen equals

$$= |\phi(k)|^2$$

from $\sum_j |\phi_k|^2 = 1$ because of quantum evolution.⁵

⁵ Choice probability for multidimensional (degenerate) projectors requires a more complex winner take all system. We leave the details for this more complex case for future research.

5.3. State reduction

The final step of the neural network implementation requires the generation of the reduced state immediately after observing r_k , which equals $\frac{\phi(r_k)}{\sqrt{p(r_k)}}$. Recall that the projectors for the confidence measurement are one dimensional (non degenerate) projectors. Therefore, if response r_k is chosen, then the reduced state is simply $\frac{P(r_k)\psi(t_1)}{\sqrt{p(r_k)}} = [0 \cdots 1 \cdots 0]^\dagger$, where there is a one located in position corresponding to the chosen response r_1 . This state reduction can be implemented in a neural network by using the single winner of the race among accumulators to form a vector of neural nodes with one active node corresponding to the winner, and all remaining nodes set to zero.⁶

6. Future extensions and final comments

6.1. Future extensions

The development presented here was based on the example of a sequence of confidence judgments made at two different time points during a signal detection task. The model for this task used the same basis for representing confidence at different time points. However, the development is more general, and it can be applied to situations that require changing the basis representing different measurements. Consider for example, the probability of a judgment about two incompatible events, A and then B (see e.g., Busemeyer et al., 2011). For example, a judge could be provided evidence about a criminal case and then asked a pair of questions: event A might represent an answer to the question “is the defendant guilty,” and event B might represent an answer to the question “should the defendant be punished by death.” Define P_A as the projector for event A and P_B as the projector for event B , and suppose the events are incompatible so that $P_A P_B \neq P_B P_A$. In this case, we need to change the basis used to represent each event. Each projector can be decomposed into a unitary matrix U of eigenvectors and a diagonal matrix Λ of eigenvalues (with zero or one values, exclusively), so that $P_A = U_A \Lambda_A U_A^\dagger$ and $P_B = U_B \Lambda_B U_B^\dagger$. The probability of A and then B equals

$$\begin{aligned} p(A, B) &= \left\| \left(U_B \Lambda_B U_B^\dagger \right) \left(U_A \Lambda_A U_A^\dagger \right) \psi \right\|^2 \\ &= \left\| \Lambda_B U_{BA} \Lambda_A U_A^\dagger \psi \right\|^2 \end{aligned}$$

where $U_{BA} = U_B^\dagger U_A$ is a unitary operator that transforms the coordinates from the A to the B basis. Therefore this computation involves unitary transformation followed by measurement in essentially the same manner as required for the sequence of two confidence ratings (see Equation (1)).

If the events in the preceding example were compatible, so that $P(A)P(B) = P(B)P(A)$, then we would need to form a tensor product space to represent the conjunctions of events: $A \cap B$, $A \cap \bar{B}$, $\bar{A} \cap B$, $\bar{A} \cap \bar{B}$. Our example, which was based on a sequence of confidence ratings, did not require the use of a tensor product space. However, once the state and the projectors are defined for a tensor product space, then the preceding development once again applies. This is because the basic principles for unitary evolution and measurement of the state are the same for tensor product spaces.

⁶ State reduction for multidimensional (degenerate) projectors requires a more complex normalization process. We leave the details for this more complex case for future research.

The most important limitation of the neural network implementation presented here is that the projectors are assumed to be one dimensional (non-degenerate). Additional work is needed on the choice probability and state reduction parts of the development to implement more general measurements that are multidimensional (degenerate).

6.2. Final comments

In the beginning of the article, we noted that researchers in the field of quantum cognition do not rely on the idea that the brain is some type of quantum computer. Instead, quantum cognition researchers only make use of the mathematical principles abstracted from quantum theory and applied outside of physics to human behavior. However, this leaves open the question of how the brain would compute quantum probabilities, if not by some type of quantum brain computer? The purpose of this article was to present one possibility for a classical neural system to implement a quantum algorithm, which is a kind of existence proof. We do not claim that there is strong evidence supporting this proposal, and there may be many better ways to do this.

The main objective that we accomplished was to formulate one way to implement the evolution of the quantum state according to unitary evolution by a neural network and then compute the probability of a response from the network that agrees with the quantum rules. We accomplished this by postulating a neural network that includes neural oscillators, which are used to generate the phases produced by unitary evolution. The phases are important for producing interference effects (violations of the classic law of total probability), which we often observe in human judgment and decision making studies (see also de Barros and Suppes, 2009). One key assumption that we made for this neural implementation is that we view this as a macro level neural model, which describes the real valued membrane potentials rather than the non negative firing rate of a pool of neurons. We assume that the firing rate is decoded back into a membrane potential at the receiving end of a pool of neurons.

Appendix

Below is the Matlab code used to compute squared magnitude from a real valued amplitude. The parameter vector $\text{parm} = [0.0852 \ 1.9564 \ 12.6635 - 12.6209]$ produces Fig. 4.

```
function [ SSE, YM ] = Pnet( parm )
% function [ SSE, YM ] = Pnet( parm )
% Fit Peter's network to squared probabilities
n=2;
p = 0:.01:1; p = p'; m = size(p,1);
a = parm(1); b = parm(2); w1 = parm(3); w2 = parm(4); T = 100;
W = [w1 w2; w2 w1];
YM = zeros(m,1);
for j=1:m
    x = p(j);
    X = [ x -x ]';
    v = .01*ones(2,1);
    for t = 1:T
        Y = Logist(v);
        dv = a.*v.*(b*v - W*Y + X );
        v = v + dv;
    end
    YM(j)=sum(v);
end
SSE = (p.^2-YM)*(p.^2-YM);
```

References

- Ashtiani, M., Azgomi, M.A., 2015. A survey of quantum-like approaches to decision making and cognition. *Math. Soc. Sci.* 75, 49–80.
- Atmanspacher, H., Filk, T., 2010. A proposed test of temporal nonlocality in bistable perception. *J. Math. Psychol.* 54, 314–321.
- De Barros, J.A., 2012. Quantum-like model of behavioral response computation using neural oscillators. *Biosystems* 110 (3), 171–182.
- de Barros, J.A., Suppes, P., 2009. Quantum mechanics, interference, and the brain. *J. Math. Psychol.* 53, 306–313.
- Busemeyer, J.R., Bruza, P.D., 2012. *Quantum Models of Cognition and Decision*. Cambridge University Press.
- Busemeyer, J.R., Wang, Z., Townsend, J., 2006. Quantum dynamics of human decision making. *J. Math. Psychol.* 50 (3), 220–241.
- Busemeyer, J.R., Wang, Z., Lambert-Mogiliansky, A., 2009. Empirical comparison of markov and quantum models of decision making. *J. Math. Psychol.* 53 (5), 423–433.
- Busemeyer, J.R., Pothos, E.M., Franco, R., Trueblood, J.S., 2011. A quantum theoretical explanation for probability judgment errors. *Psychol. Rev.* 118 (2), 193–218.
- Deco, G., Jirsa, V.K., Robinson, P.A., Breakspear, M., Friston, K., 2008. The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS Comput. Biol.* 4 (8), e1000092.
- Dirac, P.A.M., 1930/1958. *The Principles of Quantum Mechanics*. Oxford University Press.
- Eliasmith, C., 2013. *How to Build a Brain: a Neural Architecture for Biological Cognition*. Oxford University Press.
- Freeman, W.J., Vitiello, G., 2006. Nonlinear brain dynamics as macroscopic manifestation of underlying many-body dynamics. *Phys. life Rev.* 3, 93–118.
- Fries, P., 2009. Neural gamma-band synchronization as a fundamental process in cortical computation. *Annu. Rev. Neurosci.* 32, 209–224.
- Gleason, A.M., 1957. Measures on the closed subspaces of a hilbert space. *J. Math. Mech.* 6, 885–893.
- Graben, P. b, Atmanspacher, H., 2006. Complementarity in classical dynamical systems. *Found. Phys.* 36, 291–306.
- Grossberg, S., Versace, M., 2008. Spikes, synchrony, and attentive learning by laminar thalamocortical circuits. *Brain Res.* 1218, 278–312.
- Hammeroff, S.R., 1998. Quantum computation in brain microtubules? the penrose - hameroff "orch or" model of consciousness. *Philos. Trans. R. Soc. Lond. (A)* 356, 1869–1896.
- Jibu, M., Yasue, K., 1995. *Quantum Brain Dynamics and Consciousness*. Benjamins, Amsterdam.
- Kahana, M., Seelig, D., Madson, J.R., 2001. Theta returns. *Curr. Opin. Neurobiol.* 11, 739–744.
- Kak, S.C., 1995. On quantum neural computing. *Inf. Sci.* 83, 143–160.
- Khrennikov, A.Y., 2010. *Ubiquitous Quantum Structure: from Psychology to Finance*. Springer.
- Kolmogorov, A.N., 1933/1950. *Foundations of the Theory of Probability*. Chelsea Publishing Co, N.Y.
- Kvam, P.D., Pleskac, T.J., Yu, S., Busemeyer, J.R., 2015. Interference effects of choice on confidence. *112(34). Proc. Natl. Acad. Sci.* 112 (34), 10645–10650.
- Marley, A.A.J., Colonius, H., 1992. The horse race random utility model for choice probabilities and reaction times, and its comping risks interpretation. *J. Math. Psychol.* 36 (1), 1–20.
- Von Neumann, J., 1932/1955. *Mathematical Foundations of Quantum Theory*. Princeton University Press.
- Nielsen, M., Chuang, I., 2000. *Quantum Computation and Quantum Information*. Cambridge University Press.
- Nunez, P., Srinivasan, R., 2006. *Electric Fields of the Brain: the Neurophysics of Eeg*, second ed. Oxford University Press.
- O'Reilly, R.C., Munakata, Y., Frank, M.J., Hazy, T.E., Contributors, 2012. *Computational Cognitive Neuroscience*, first ed. Wiki Book <http://ccnbook.colorado.edu>.
- Pitowski, I., 2006. Quantum mechanics as a theory of probability. In: Demopoulos, W., P., I. (Eds.), *Physical Theory and its Interpretation: Essays in Honor of Jeffrey Bub*. Springer, Dordrecht, pp. 213–239.
- Pleskac, T.J., Busemeyer, J.R., 2010. Two-stage dynamic signal detection: a theory of choice, decision time, and confidence. *Psychol. Rev.* 117 (3), 864–901.
- Pothos, E.M., Busemeyer, J.R., 2009. A quantum probability model explanation for violations of 'rational' decision making. *Proc. R. Soc. B* 276 (1665), 2171–2178.
- Ratcliff, R., Smith, P., 2004. A comparison of sequential sampling models for two-choice reaction time. *Psychol. Rev.* 111, 333–367.
- Shadlen, M.N., Movshon, J.A., 1999. Synchrony unbound: a critical evaluation of the temporal binding hypothesis. *Neuron* 24, 66–77.
- Stewart, T.C., Eliasmith, C., 2013. Realistic neurons can compute the operations needed by quantum probability theory and other vector symbolic structures. *Behav. Brain Sci.* 36 (3), 307–308.
- Suppes, P., 1966. The probabilistic argument for a nonclassical logic in quantum mechanics. *Philos. Sci.* 33, 14–21.
- Takahashi, T., Cheon, T., 2012. A nonlinear neural population coding theory of quantum cognition and decision making. *World J. Neurosci.* 2, 183–186.
- Verdonck, S., Tuerlinckx, F., 2014. The ising decision maker: a binary stochastic network for choice response time. *Psychol. Rev.* 121 (3), 422.
- Wang, Z., Busemeyer, J.R., 2016. Interference effects of categorization on decision making. *Cognition* 150, 133–149.
- Ward, L.M., 2003. Synchronous neural oscillations and cognitive processes. *Trends Cognitive Sci.* 7 (12), 553–559.