Connection Science
Vol. 00, No. 00, Month 2011, 1–14

RESEARCH ARTICLE

Preventing combinatorial explosion in a localist, neural network architecture using temporal synchrony

Patrick Simen[a]*

[a]*Princeton Neuroscience Institute, Green Hall, Princeton, NJ, 08544*;
(*Received 00 Month 200x; final version received 00 Month 200x*)

I evaluate the pros and cons of a cognitive modeling approach based strictly on localist representations by examining a sequential decision making architecture developed to model the neural basis of problem solving. This architecture, comprising layers of graded-activation processing units interleaved with layers of approximately binary units, seems not only amenable to distributed representations in its graded layers, but actively in need of them in order to support levels of model complexity on the scale of symbolic systems. The architecture employs a key connectionist principle: that the semantics of a representation are defined by what is connected to what. Because of this choice, however, the architecture requires a combinatorially explosive number of localist units as problem complexity increases. I discuss the possibility of preventing combinatorial explosion by binding low-level representations into high-level representations through temporal synchrony, using the same dynamics that underlie decision making in the architecture.

Keywords: Production system; neural network; diffusion model; binding; synchrony; localist

1. Introduction

The distributed/localist dimension in the space of cognitive architectures parallels the subsymbolic/symbolic dimension, in which the choice is between graded and discrete numerical representations, respectively. In this paper, I describe a localist architecture that converts subsymbolic representations into symbolic ones during problem solving. Solving a problem in this architecture reduces to making a sequence of atomic decisions about where to go next in a problem space; each decision converts graded evidence about where to go into an all-or-none decision about the next node to visit in the problem space. I demonstrate that this architecture, which accounts for interesting patterns of problem solving deficits in patients with prefrontal brain damage Polk et al. (2002) and Parkinson's disease Simen et al. (2004), nevertheless suffers from extreme inefficiency in the use of neural resources: The need for unique localist units to code for conjunctions of other active units grows combinatorially as problem complexity increases.

The psychological and neuroscientific study of perceptual decision making suggests alternative representational mechanisms, however. Such research usually focuses on the translation of sensory information — which is represented in primary sensory cortex in a presumably subsymbolic and distributed form — into discrete motor actions, which (I presume) are mediated by symbolic/localist representations

*Corresponding author. Email: psimen@math.princeton.edu

ISSN: 0954-0091 print/ISSN 1360-0494 online

DOI: 10.1080/09540090xxxxxxxxxxxx
http://www.informaworld.com

in primary or supplementary motor cortex. Much work in this area also suggests a role for synchronous oscillations across distant brain regions during decision making. I therefore conclude the paper by examining a particular implementation of binding-by-synchrony (one approach to solving the 'binding problem'). This implementation may allow sufficient control over the time at which component representations are active to build temporary conjunctions of localist representations, and thereby avoid combinatorial explosion without sacrificing functionality. Most importantly, it supports a form of self-organization in time: distinct concepts that share components are by definition in conflict with each other, and this conflict triggers a decision making process based on lateral inhibition to resolve it. This resolution splits time into phases during which only one of the conflicting concepts is active. The organization of *which* concepts are active *when*, however, is determined entirely by interactions between the concepts themselves. This obviates any need for an external controller/sequencer somewhere else, and thus averts the danger of an infinite regress of controllers.

2. Natural and artificial decisions, algorithms and automata

The power of the Turing machine architecture for computation makes it almost irresistible as a framework in which to create cognitive models. Different physical models of computation — e.g., the mind as an hydraulic system of pipes, pumps and reservoirs, represented by differential equations — can themselves be emulated using scientific computation techniques on a standard computer. Modeling cognition in this way requires more work than simply programming with propositional logic, however; and if one's goal is a model of problem solving, for example, the payoff is often not great. Thus a strong emphasis on circuit-level descriptions of cognitive architecture components strikes many AI researchers and some cognitive psychologists as getting everything precisely backwards. Computer technology allows a complete separation of levels of description: the physical level, involving transistors and resistors, can be completely ignored by circuit designers who combine off-the-shelf logic gates and clocks to design embedded circuits at the logic level. Software engineers similarly benefit from never having to think about instruction pipeline characteristics in a CPU. Instead, they are able to focus on the computational or algorithmic problem at hand, to use Marr's terminology. If they ever do have to move down a level (say, to speed up their code), then it is feasible, but rarely necessary, to do so.

The ability to create symbolic cognitive architectures like ACT-R and Soar similarly depends on the existence of an underlying platform that cleanly separates levels of description. This standard computing platform, with its sequential reading of a working memory buffer and algorithmic selection of rules, incorporates a number of features that seem implausible as models of the circuit-level structure of the brain, but that invisibly provide a great deal of power to the cognitive modeler. (In fairness, recent work with ACT-R focuses more closely on circuit-level descriptions, but low-level assumptions of sequential circuit design in digital electronics still seem to be implicit in this work; cf. Stocco et al. 2010). More importantly, the failure to date of AI to produce resilient, autonomous agents may (I conjecture) derive from ignoring the fact that in the transition from animals like mice into animals like humans, Turing-machine-like abilities emerged from a circuit structure in which a clean separation between levels of description is likely impossible.

I will therefore sketch out my attempt to take the physical, circuit-level description of the brain seriously while specifying design principles for an architecture capable of symbolic processing in the form of problem space search. This architecture

will appear distributed, stochastic and continuous at the lowest level of description, and approximately localist, deterministic and discrete at a higher level. The mathematical language used to describe the lowest level will consist of stochastic differential equations, while propositional logic will often suffice for describing the highest level. The cognitive modeling sweet spot, I conjecture, consists of a marriage of continuous, stochastic, subsymbolic process descriptions, via decision making, with the modular, compositional, hierarchical structures that have been the bread and butter of AI research. The result will unify random walk/diffusion models of decision making, neural networks, and capacity-limited production systems, in a system that requires multiple levels of description: at the lowest level, in terms of analog, asynchronous, stochastic processes guided by Hebbian and error-driven learning combined with symbolic; at the highest level, in terms of rule-driven processes guided by heuristic problem space search. Ironically, this marriage would actually unite two distinct forms of decision making that Turing himself worked on in separate contexts: 1) mathematician David Hilbert's decision problem, which was to determine whether an algorithm could be devised for automatically deciding the truth or falsehood of any statement in a given mathematical language (answer: no), and 2) deciding among hypotheses in the face of sequential, noisy samples of evidence (specifically, cracking the German navy's Enigma code in World War II; see Gold and Shadlen 2002). Actually completing this unification of modeling techniques will founder, as other attempts have in the past, on the binding problem, in which it becomes problematic to assign simultaneously active representational components to the appropriate representations in a parallel processing system. The last part of the paper will discuss a mechanism for solving the binding problem through temporal synchrony, which seems promising but remains work in progress.

3. Noise, evidence accumulation and thresholds

Finite state automata and regular expressions were proposed as a model of neural processing by Kleene (1956), just as behaviorism's dominance began to wane in psychology. Internal psychological states, unobservable as they appeared to be, were considered by many psychologists at the time to be unworthy subjects of study. Kleene's state-preserving automata are equivalent to the essential control device at the heart of every Turing machine, however, and are thus central to the Turing machine's capacity to capture something essential about human thought. I claim that neural circuits, even in simpler organisms like mice and flies, must implement finite state automata in order to support a suite of qualitatively distinct behaviors, each of which can be appropriately triggered by environmental stimuli. Nevertheless, they need not do so in the way that modern digital technology implements them in computers, with synchronous circuit updates, binary voltage levels, low noise and a central system clock.

Instead, I propose a process of statistical hypothesis testing, with graded levels of evidence that accumulates continuously over time, finally triggering the crossing into one or another distinct state at some critical level of evidence (cf. Kopecz and Schoner 1995; Usher and McClelland 2001). Such a process corresponds to passing a bifurcation point in a dynamical system: a 'point of no return' for the underlying continuous system that makes it appear digital and discrete when viewed macroscopically. An analogous switching process occurs in every digital, synchronous circuit, but I envision a process that is more subject to noise, uses no clock signal, and is based on a linear superposition of evidence in favor of the various destination states that can be reached from the current state. I propose that the transition from graded levels of evidence into distinct states defines the transition from a

subsymbolic to symbolic representational scheme, and furthermore may represent a transition from distributed to localist representations in a system. What particularly distinguishes this approach from modern computing architectures is that the evidence accumulation process can be quite extended in time, with rich dynamics driven by circuit structures acquired from experience by statistical learning procedures, such as back-propagation or Hebbian learning.

A mechanism for implementing threshold-crossing detection is critical to this endeavor. A simple strategy is to emulate the threshold mechanism that generates action potentials in the individual neural membrane: below a critical axonal membrane potential, the potential is attracted toward a hyperpolarized state in the absence of excitatory impulses; above a critical level, voltage-gated ion channels suddenly reverse the membrane potential's attractor to a much higher value, after which a second current again reverses the level of the attractor to the hyperpolarized potential. In principle, then, all of the subsymbolic processing under discussion could take place in the dynamics of sub-threshold membrane potentials, and decisions could be made by the firing of a single action potential. On a cognitive time scale, however, such a scheme seems unlikely to work and, in any case, appears unsupported by physiological evidence. Instead, a firing-rate code seems to be used for making difficult perceptual decisions, and a classic model of the firing rates of neural populations as simple nonlinear filters — classic neural network units — seems more likely to supply the neural code for cognition (at any rate, the neural code for processes that evolve over time scales much longer than the 10 msec timescale of the neural membrane).

Let us suppose then that a simple, sigmoidal activation function describes the input/output relationship of a neural population (leaving noise out of the picture for the moment), so that populations act like leaky integrators, or active, low-pass filters: i.e., capacitors connected to operational amplifiers with a (mostly) linear gain that falls off at very high input levels. Leaky integrators that excite themselves via recurrent excitation, and that strike a perfect balance between leak and recurrent excitation, can act as perfect integrators over a non-negligible range of inputs. If self-excitation is turned up beyond this critical level, such a population exhibits bistability and hysteresis. Simply put, it acts like a *switch*, with only two, widely separated levels of stable activation. More importantly, once it transitions from one state to the other, it tends to stay there, until forced back to the other state by a sustained change in inputs. In contrast, ideal switches that act as perfect step functions, or Heaviside functions, produce disastrous chatter in noisy environments, transitioning from off to on and back arbitrarily quickly as inputs fluctuate. In another important respect, this model of a neural switch differs from the CMOS switches widely used in digital electronics in that it still displays an appreciable amount of graded activation: although there exists a range of mid-level output values that are transient and result in transition into 'up' or 'down' states, the activation levels lumped into these non-contiguous states nonetheless show graded changes in response to changes at all input levels other than those defining the system's bifurcation points. The switch is thus like an unusual dimmer control for a light bulb — one that refuses to stay put within an intermediate range of settings, so that the room can either range from pitch black to very dim, or from very bright to blinding.

I and my colleagues have previously attached the outputs of a neural network implementation of a drift-diffusion or Ornstein-Uhlenbeck model of decision making to the inputs of a set of such switch mechanisms to obtain a complete decision making model Simen and Cohen (2009). Without the energy barriers imposed by switches, evidence accumulation in a decision making circuit would propagate

through the system to motor actuators and produce graded levels of movement. Switches hide the state of evidence accumulation from the world, allowing it out only in an approximately punctate burst, like an action potential. With this model of a threshold in hand, schemes for adapting speed-accuracy tradeoffs and response biases via threshold adaptation become quite simple (e.g., Simen et al. 2006).

Most importantly, such bistable mechanisms allow relaxation oscillators to be constructed that will underlie the temporal synchrony mechanism alluded to previously, as well as a switching architecture based on local, hand-shake procedures that carry out complex, sequential behavior without the aid of a central system clock.

4. Neural population model

The basic building block I propose is a stochastic neural network unit. I begin its description by considering it as a deterministic system. At each moment, it computes a weighted sum of its current inputs, then computes an exponentially decaying average of recent weighted sums, and finally amplifies the result by a gain function that is approximately linear (but which saturates at very low and very high input levels). This quantity is broadcast to other units, over connections whose strengths determine their relative contribution in those units' weighted sum computations. Formally, the output of the ith unit is V_i, the leaky integral of summed input is x_i, and the dynamics are defined as follows:

$$I_i = \sum_{j=1}^{n} w_{ij} \cdot V_j, \tag{1}$$

$$\tau \cdot \frac{dx_i}{dt} = -x_i + I_i, \tag{2}$$

$$\text{and} \quad V_i(t) = f(x_i(t)) = [1 + \exp(-\lambda \cdot (x_i - \beta))]^{-1}. \tag{3}$$

Parameters λ and β determine the steepness and position of the sigmoidal activation function f, and τ determines the decay rate of exponential averaging (large τ gives slow decay).

In addition to deterministic dynamics, I assume that noise enters the system from units that have direct sensory inputs, and also from the units themselves. To model these assumptions, I use stochastic differential equations, in which I represent white noise with a useful abuse of notation as $\eta \equiv dW/dt$ (multiplication by dt then gives the standard notation dW in our equations; cf. Gardiner 2004). This quantity represents the time-derivative of a Brownian motion, or Wiener process, $W(t)$.[1] The standard deviation of η is 1, but can be changed to any value c by multiplying by c. Here, we multiply η by the square root of the weighted input, an assumption which is consistent with an even more microscopic level of neural modeling: I assume that spiking neurons are Poisson processes, and that leaky integrators model their population-level behavior (cf. Smith 2010). The variance of sums of these independent processes is the sum of their variances. Thus we get a noise standard deviation proportional to the square root of net input, with the proportionality constant depending on the weights w_{ij}. Formally, then, the full,

[1] W in fact is non-differentiable, but it is the limit of a sequence of slightly smoother, differentiable noise processes, so it can be used without danger.

stochastic unit description is as follows:

$$\begin{aligned} \tau \cdot \frac{dx_i}{dt} &= -x_i + \left(I_i + c_{ij}\sqrt{I_i} \cdot \eta\right) \\ \Rightarrow \tau \cdot dx_i &= (-x_i + I_i)\ dt\ + c_{ij}\sqrt{I_i}\ dW_{ij} \\ \Rightarrow \tau \cdot dV_i &\approx (-x_i + f(I_i))\ dt\ + c_{ij}\sqrt{I_i}\ dW_{ij} \end{aligned} \tag{4}$$

(See Simen and Polk (2010) for justification of the last approximation, which moves the noise term outside the nonlinear function f.)

This system can be numerically simulated on a computer (and perhaps be more easily understood) as a discrete-time difference equation Gardiner (2004):

$$\tau \cdot V_i(t + \Delta t) \approx V_i(t) + (-x_i + f(I_i))\ \Delta t\ + c_{ij}\sqrt{I_i}\sqrt{\Delta t}. \tag{5}$$

It is now critical for our purposes to consider the effects of recurrent excitation of a unit by itself ($w_{ii} > 0$). The strength of this self-excitation determines which of three, qualitatively distinct types of behavior a unit exhibits Simen and Polk (2010). For $w_{ii} < 1$, the system acts like a leaky integrator; as w_{ii} grows, the leak is reduced. When the self-excitation exactly balances the leak ($w_{ii} = 1$), the unit acts like a perfect integrator (until it saturates). For $w_{ii} > 1$, the system is unstable and is forced upward against the upper ceiling on its activation (1), or downward toward its lower floor (0); thus it acts like a binary switch. Furthermore, such a unit displays hysteresis, so that it can both trigger abrupt changes and also store a bit. Fig. 1 shows the dynamics of such a unit. Equilibrium curves in Fig. 1**a** and **b** are solid; velocities dV/dt are indicated by arrows and shading (light > 0, dark < 0).

In general, leaky integration (weak self-excitation, indicated by a sigmoid symbol in Fig. 1**c**) is useful because it low-pass filters its input, thereby removing much of the high frequency noise contributed by noisy spiking and by the environment. Perfect integration (balanced self-excitation, indicated by a rounded step-function symbol in Fig. 1**c**) is needed for optimal hypothesis testing Bogacz et al. (2006). Bistability (strong self-excitation, indicated by an S-shaped symbol in 1**c**) is needed for triggering subsequent steps of sequential processes and for maintaining the current state of working memory. Bistable units act as latches in digital electronics and can store a 1 (upper gray region of Fig. 1**d**) or a 0 (lower gray region of Fig. 1**d**) as long as input is held between A and B. This is because states above the dashed curve converge to the upper solid curve, while states below it converge to the lower solid curve. Bit-flipping during constant I is least likely when $I = (A + B)/2$.

5. Neural productions, timers and oscillators

Fig. 2 shows the basic building blocks of the proposed architecture; in the remainder of the paper, I explain how each block functions, and show how the localist assumptions inherent in them allow for controllable sequential processing, but suffer from a curse of dimensionality. The left column shows the 3 unit types (**a**,**b**,**c**). Simen and Polk (2010) detail how a complete set of logic operations (AND, OR, NOT) can be built from the bistable units in **c** by parameterizing their input strengths. Panel **d** shows a simple if-then rule structure: the leaky integrator filters noise from its inputs, and if the sum exceeds a critical level, the bistable unit switches from (approximately) 0 to (approximately) 1. This is analogous to the process of

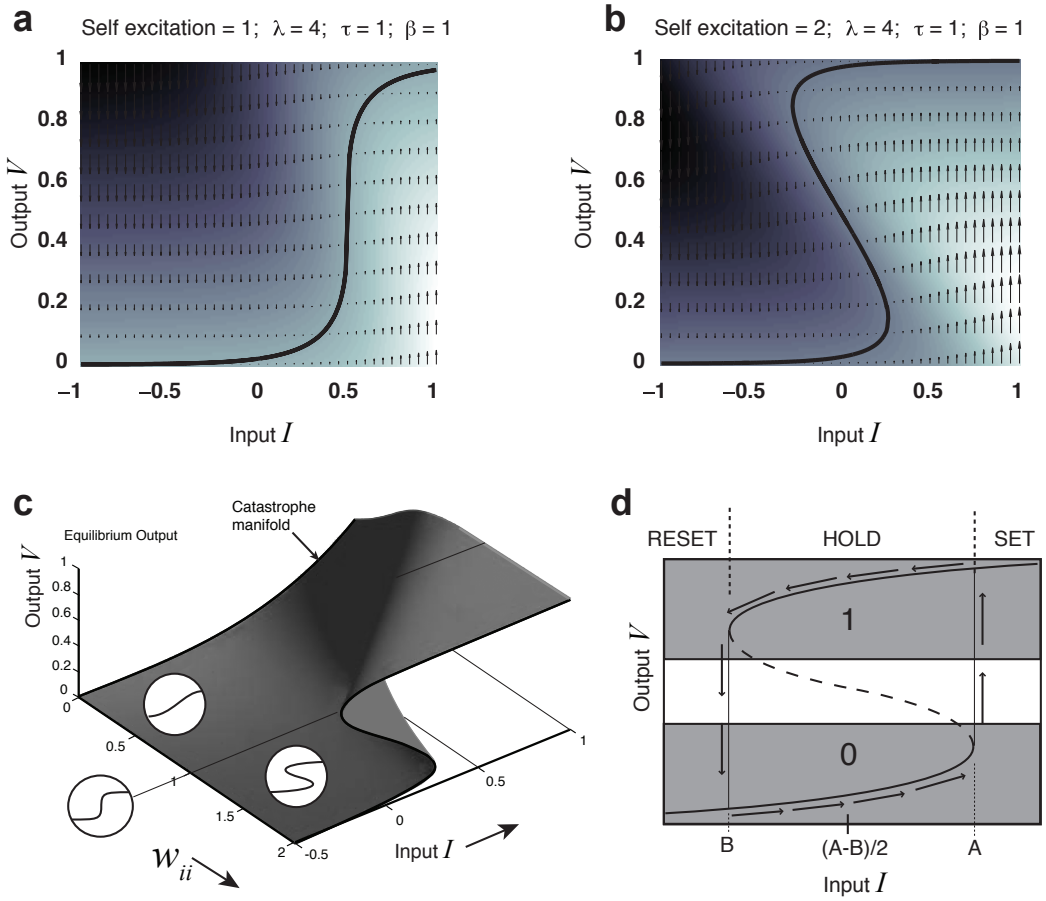

Figure 1. **a, b:** A neural network unit's rate of activation change (dV/dt) as a function of input I and output V for units with fixed I and balanced (**a**) or strong (**b**) excitatory, recurrent connections. **c:** 'Catastrophe manifold' formed by the equilibrium curves of Eq. 3 as the self-excitatory, recurrent weight strength w_{ii} ranges from 0 to 2. **d:** A latch based on hysteresis.

'matching' the contents of working memory (which can be made to depend on arbitrarily many symbolic preconditions using a cascade of logic gates). The degree of match may be an analog quantity, and whether this is sufficient to cause a bit flip in the output unit determines whether the 'production', or if-then rule, will 'fire' Newell (1990). Furthermore, the weights on inputs to the if-stage may also encode preferences between productions that have an equal degree of supporting evidence.

If more than one production matches, however, there may be conflict between them. At least at the motor output stage (e.g., SOAR's 'operators'), such conflict must be resolved. Here I consider conflict resolution as a process of competition between matching productions (Fig. 2 **e**), with the outcome biased toward selection of the production with the strongest amount of preference-weighted evidence. Since noise is everywhere, this reduces to a well-defined hypothesis testing problem, for which simple, near-optimal algorithms exist. These algorithms — sequential probability ratio tests (SPRTs) — can be parameterized to maximize expected utility in the case of two-alternative choices Bogacz et al. (2006), and can approximately maximize utility for a greater number of competing alternatives McMillen and Holmes (2006), Simen et al. (2010). For a difficult decision, the process of deciding via lateral inhibition (a form of attractor dynamics) can be parameterized to implement an SPRT. This requires only that the lateral inhibitory strengths between input units equal -1. An example of these dynamics is shown in Fig. 3. There, the

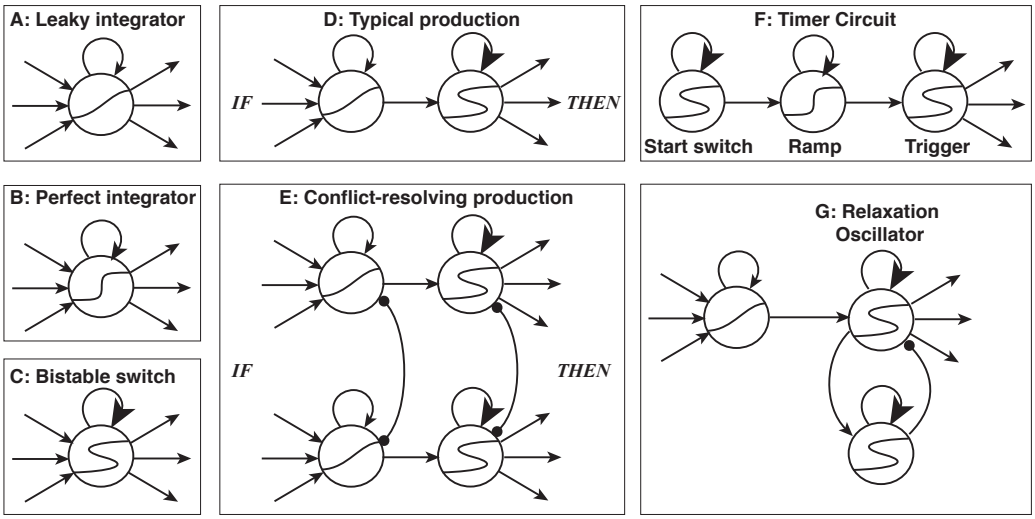

Figure 2. Basic building blocks. Arrowheads indicate excitation, circleheads inhibition. **a, b, c:** Elementary particles; arrows: excitatory inputs. **d:** Production topology. **e:** Conflict resolution via lateral inhibition (circles: inhibition); inhibition between switches is optional. **f:** Interval timer. **g:** Relaxation oscillator added to production output unit.

2D system in the bottom layer of units reduces to a single dimension (Fig. 3**c**), along which a random walk to threshold occurs. The threshold is implemented by attractor dynamics in the top layer of units, the dynamics of which are shown in Fig. 3**d**. Thus, the firing of a single production is equivalent to a statistical hypothesis test.

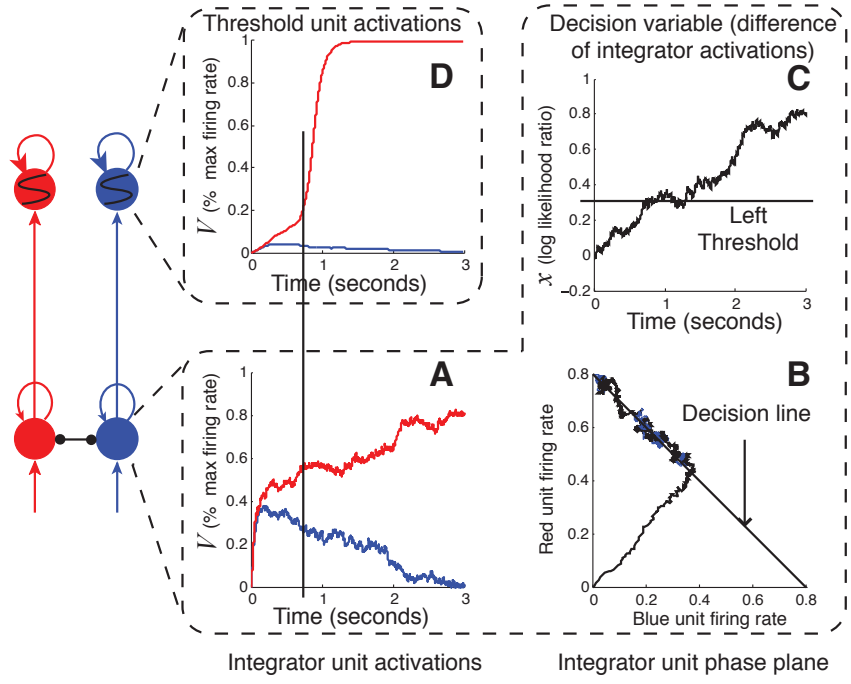

Figure 3. Hypothesis testing via lateral inhibition, equivalent to a conflict-resolving production (Fig. 2**e**). **a:** Activation of each evidence accumulator in the bottom layer of units. **b:** Same activation, depicted in a phase plane format (red-unit activation plotted vs. blue-unit activation). **c:** Random walk representation. **d:** Threshold unit activations over time.

A critical question facing the proposed architecture, however, is whether the

timing of these firings can be coordinated and sequentialized without reference to a central system clock. Our problem is the same as that facing digital circuit designers, who have long relied on a central clock and synchronous updating to preclude critical race conditions and other signal timing hazards. Our solution is to use these production implementations to form processing bottlenecks, and to use handshake completion signals between computing elements for asynchronous, distributed timing control Simen and Polk (2010). The most difficult question is whether we can implement productions of the form: If A, Then B and Not A. Naively wiring up a system to implement such a production can cause critical race conditions or metastability.

Our solution derives from the hysteresis properties of our bistable units. Fig. 4 shows that a sequence of such units can be wired up so that an input unit stays active long enough to trigger an output unit, which in turn inhibits the input. If the input unit did not resist this inhibition, it could fail to latch the output before shutting off. Elsewhere I have detailed the specific conditions that ensure proper sequential latching Simen and Polk (2010). To ensure that timing issues can be handled, I use the timer circuit in Fig. 2 **f** to implement an analogue of the delay gates used in digital logic. This mechanism activates a 'start' switch unit on the left, then integrates that signal in a 'ramp' unit, weighted by the start-to-ramp weight, until it triggers the 'trigger' unit to flip from 0 to 1. The delay duration is equal to this threshold value divided by the start-to-ramp weight. These dynamics are very similar to those implementing hypothesis-testing in Fig. 3, but now the only evidence is the passing of time.

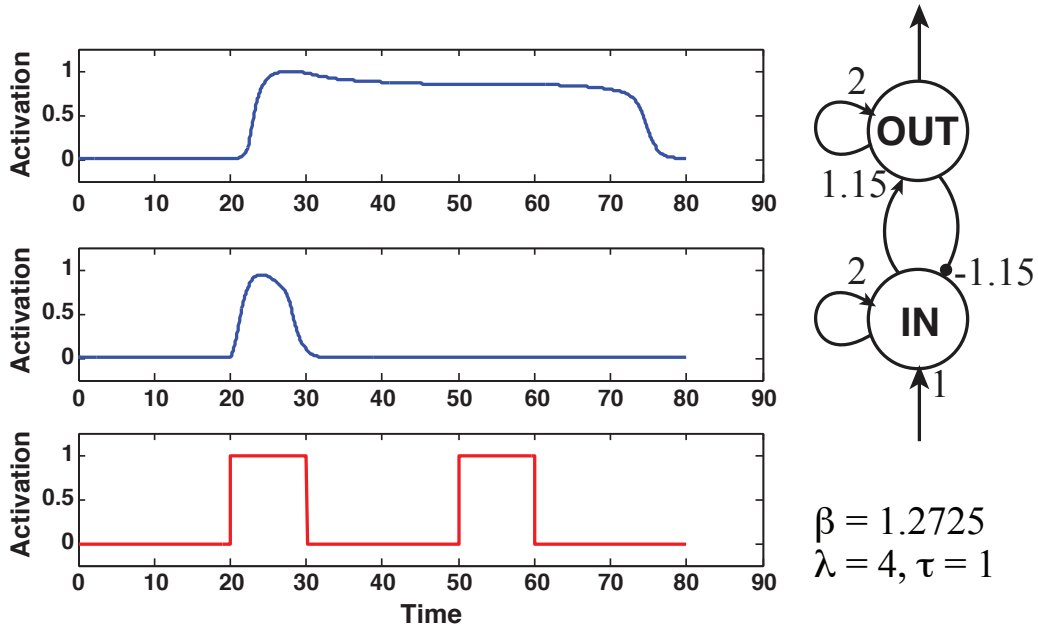

Figure 4. A production that negates its own if-condition. Bottom layer: input signal (red). Middle layer: IN unit activation. Top layer: OUT unit activation.

With these building blocks in hand, we can build arbitrarily complex circuits that implement logic gates and finite state machines, and thus special-purpose production systems, such as the Tower of London problem solver discussed at the end of the paper. However, we still face the same critical problems facing all connectionist systems: if the semantics of a representation depend on what is connected to what, then how do separate representations share subcomponents? Or if their subcomponents conflict, then how are the proper subcomponents bound with the proper

parent representation? Temporal synchrony has been widely considered to be a potential solution. The architectural assumptions are that whatever is simultaneously active refers to the same entity, and distinct entities share different oscillation phases (cf. Hummel and Holyoak 1997). I implement these assumptions using the same machinery that underlies productions which cancel their own if-conditions.

Fig. 2 **g** shows that for each production trigger, we can assign an inhibitor. If a production fires, its output unit activates and triggers its own cancellation after a controllable delay (depending on connection strengths). However, the firing of a production can trigger a stored, hidden variable in a third bistable unit, which forces reactivation of the production after the inhibitor falls silent. This process repeats, triggering oscillations. When productions compete with each other, they push their active periods out of phase with each other, as shown in Fig. 5. When they do not, excitation causes them to entrain to the same phase. Thus conflicting representations locally decide which gets to broadcast information globally. If we allow for a plasticity signal that globally increases the learning rate of Hebbian connection plasticity between units, and if we activate this signal only at critical times, then we can burn in connections (possibly temporary connections) between units simply by activating them.

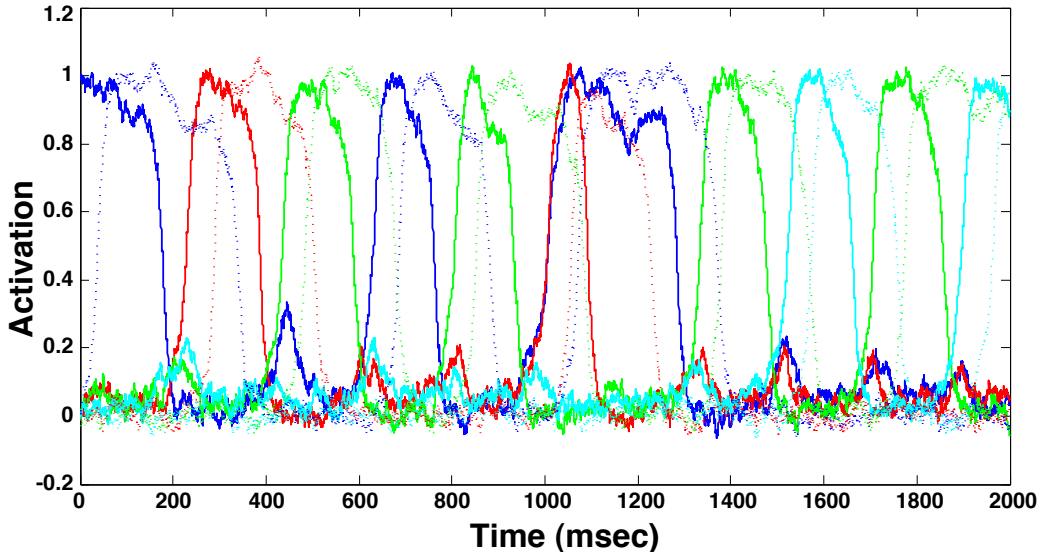

Figure 5. Relaxation oscillations among competing representations, allowing sharing of a single broadcast channel. Each solid color corresponds to one representation's bistable output unit; dashed curves correspond to the output's inhibitor.

6. Neural model of the Tower of London task

To demonstrate both the power and the weakness of the architecture currently proposed (minus its temporal synchrony generators), I now give an overview of a model of problem solving in the Tower of London task built within the architecture. This model is illustrated in abstract form in Fig. 6. Each module represented by an oval in that schematic consists of a two-layer, laterally inhibiting set of conflict-resolving productions as in Fig. 2**e**. The input layer implements a high-dimensional random-walk/diffusion process, and the second layer of switches commits the module to one choice when a critical level of evidence is accumulated in the input layer. It thereafter preserves the choice by exploiting hysteresis in the output layer of (localist) switch units. Such propositional information is preserved indefinitely this

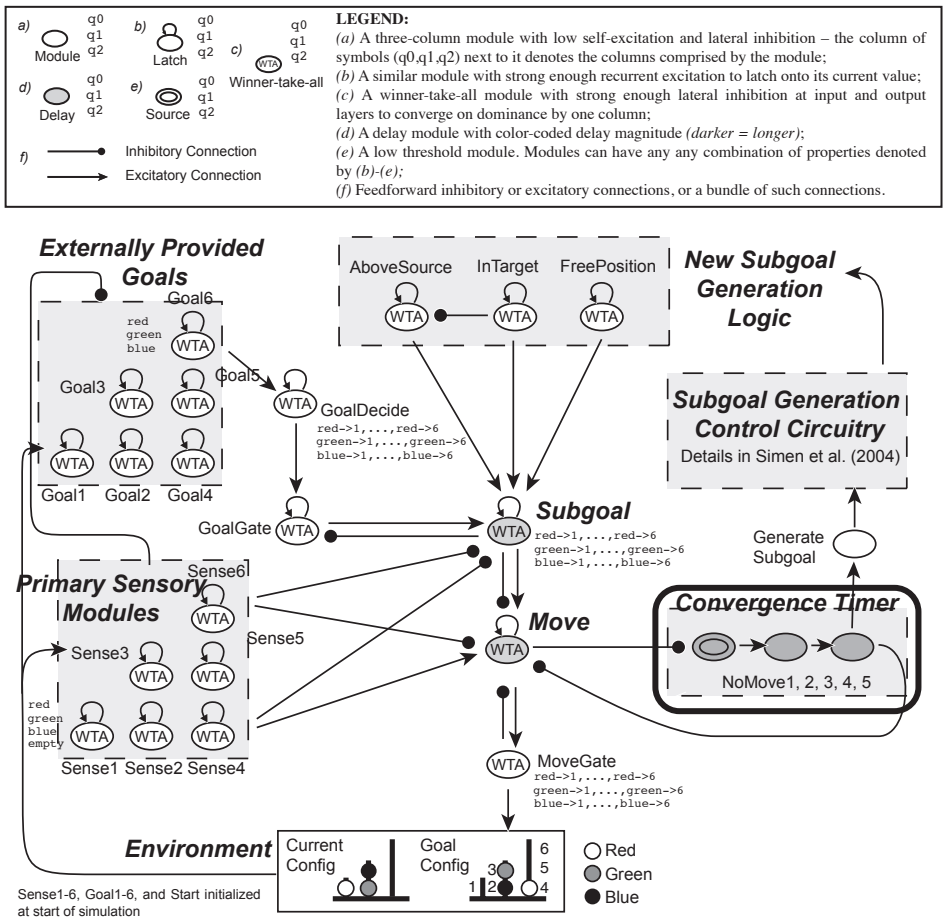

Figure 6. Tower of London model schematic. Each oval is a multi-layer module as depicted in Fig. 2**e**. The object of the game is to move the colored balls from a starting to a goal configuration in the minimum number of moves possible.

way, until noise knocks it down or until sufficiently strong inhibition from elsewhere in the model is received.

In the model problem solver, a set of Sensory modules, one for each position of the gameboard, is initialized to (localist) patterns encoding the color of a ball at that position, if any, and these representations then persist until reinitialized by changes in the environment. They in turn excite the representations of legal moves in a separate Move module devoted to action representations, and inhibit illegal ones. Diffusion/attractor dynamics within the Move module results in the selection of a single action for execution, completing the simulation of a simple production of the form: 'if the red ball is in position X, then place it in position Y'.

It is important to note how similar this choice process is to the type of perceptual decision making processes discussed in, e.g., Ratcliff and McKoon (2008) and Usher and McClelland (2001), and how different such a choice is from action selection by a game-tree search algorithm (e.g., breadth-first search; cf. Russell and Norvig 1995). It allows — indeed, demands — a random action selection time, and allows for the weighted sum of influences of a host of representations throughout the model.

6.1. *Goals and subgoals*

In the Tower of London solver, one set of winner-take-all modules is dedicated to the representation of externally defined goals and another to internally generated

subgoals. Activation in the goal modules biases the competition taking place in the Move module, favoring moves of one ball over the others. This biasing is just another form of production, but the if-condition is semantically special: it represents a desired state of the environment. Further, the biasing strength of such a production is insufficient to activate its then-condition without support from some other module, as in the case of the Sensory module just discussed. Technically, this Goal → Move excitation should be considered only a component of a rule of the form: 'if Goal is X and Percept is Y, then Do Z'.

6.2. *Model performance*

Timecourses of activation in key components of the model are shown in Fig. 7. The problem, shown at the bottom of Fig. 6, requires five moves for solution and therefore requires that some balls be moved to positions other than their final, goal positions. Thus it requires the internal generation of subgoals for efficient solution. The Sense modules, like the Goal modules, are initialized at the beginning of the simulation and excite potentially legal moves. A winner, 'Red to 4' is selected at time point A, and the corresponding unit in MoveGate is caused to rise to threshold, achieving the move and wiping out the move-generating command in Move. At this point, the simulated environment causes an update of the Sense modules, which in turn extinguish any goal or subgoal activation pattern in the Goal system or Subgoal which represent goals to create the current environmental configuration (point B). This allows the next most preferred goal to be retrieved and worked on, as can be seen in Subgoal at point C. At no point is the Convergence Timer system involved.

Now the next goal, 'Blue to 2', which is unachievable, has been selected, and this in turn generates a subgoal to remove an obstacle. Once a subgoal is selected ('Green to 5', since Green is in the target position of the blue ball, at time point D), the first element of the Convergence Timer sequence (NoMove 1) begins to ramp up, and finally maximal activation reaches the last timer in the sequence at time E (this also happens for the previous goal). This activates the Generate module for generating a subgoal. Finally, the subgoal generation logic computes that the ball above the green source ball is blue, at time F, and that the lowest position on a peg which is neither the source nor the target of the goal is position 1 at time G, and Subgoal responds to this voting at time H. The model continues on in this way until eventually solving the problem in 5 moves, as is shown in the sequence of moves selected by the model.

7. Curse of dimensionality with a localist representational scheme

The Tower of London model presented above suffers from an inability to scale up to larger problem spaces. The Move module represents moves with a set of localist units, each of which encodes one of the 18 unique pairings of a ball color and a target position for that ball. As the number of board positions increases, this scheme requires a linear increase in resources; as the number of both game pieces and positions increases, a combinatorial explosion of resource requirements occurs.

One way to combat this explosion is to combine localist representations of color (3 values) with representations of target positions (6 values) in a temporal synchrony code as previously described. Under this scheme, colors and positions active at the same time represent a given move. This amounts to a very limited move toward a distributed form of representation, but the payoff could be quite substantial.

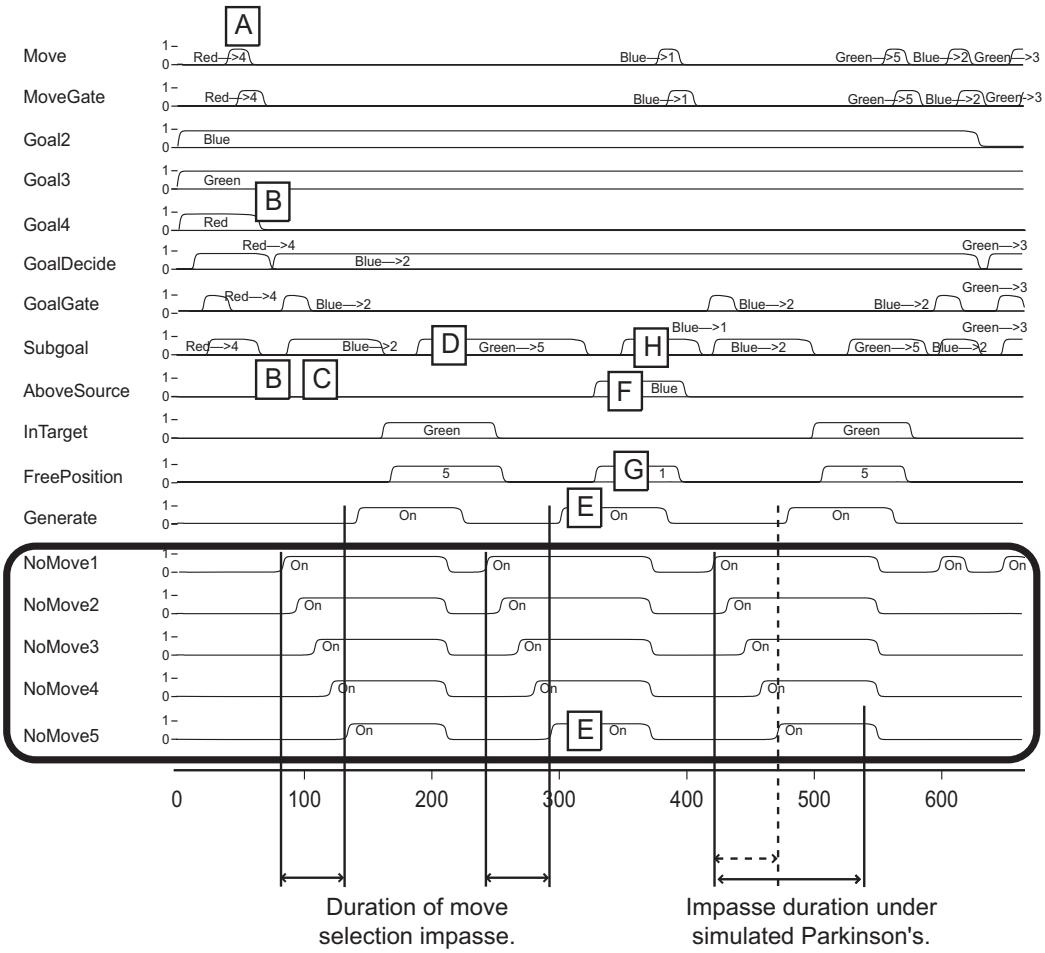

Figure 7. Time course of Output unit activation in most modules of the model during the solution of a five-move-minimum problem, depicted in the Environment panel at the bottom of Fig. 6. Delay between onset of activation in NoMove1 and NoMove5 defines the time window in which a move can be selected before a subgoal is generated. This delay increases as Input → Delay inhibition is weakened, producing the model's latency impairment.

8. Discussion

The type of localist system used here to implement finite state automata and limited production systems offers more of the type of sequential decision-making expected from AI programs than is typical in a neural network. However, nothing here should be taken to preclude the kind of subsymbolic computation that is the hallmark of parallel distributed processing models. The connection strengths in the Tower of London model were designed to trap the system into a rigid sequence of step-by-step decisions, as an exercise in showing how closely automata (i.e., memory-limited Turing machines) could be emulated, while remaining committed to fundamentally subsymbolic decision making at the model's core. Greater flexibility could be achieved by allowing the sequencing to fall apart with some probability, and it seems possible that many of the distinct localist representations could be multiplexed onto a smaller number of units.

More work would be needed to determine the scope of the proposed approach to dynamic symbol and rule creation, to the implementation of a data type system such as exists in ACT-R and Soar, and to the binding problem more generally. With enough additional assumptions about the structure of the basic building blocks, of course, it would be possible to translate between any given symbolic architecture

and an architecture built from the components we have outlined. This must be the case in a trivial sense because the components used here are equivalent to circuits of resistors, capacitors and transistors — the building blocks of modern computers. Which additional assumptions are actually justified for cognitive modeling will require a great deal of empirical research, but a program of purely theoretical exploration that focuses on the emergence of a higher level of description from a lower level seems to me to be a critical enterprise nevertheless. Attempts to unify modeling approaches across levels are certainly not destined to succeed, but even their failure is likely to be instructive.

In this case, the failure is clearly one in which a combinatorial explosion of localist representations occurs as problem complexity increases. A possible remedy exists in dynamically binding together inherently localist representations into distributed complexes whose simultaneous activation represents a given entity at a particular phase of an oscillation. These representations would then feed back into the selection of a localist representation in the process of making the next in a sequence of decisions. The processing dynamics of such an architecture thus amount to a continuous alternation between localist and distributed forms of representation — a kind of dynamics that seems worthy of further investigation.

References

Polk, T.A., Simen, P.A., Lewis, R.L., and Freedman, E.G. (2002), "A computational approach to control in complex cognition," *Cognitive Brain Research*, 15(1), 71–83.

Simen, P.A., Polk, T.A., Lewis, R.L., and Freedman, E. (2004), "A computational account of latency impairments in problem solving by Parkinson's patients," in *Proceedings of the International Conference on Cognitive Modeling*, pp. 273–279.

Stocco, A., Lebiere, C., and Anderson, J.R. (2010), "Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination," *Psychological Review*, 117, 540–574.

Gold, J.I., and Shadlen, M.N. (2002), "Banburismus and the brain: decoding the relationship between sensory stimuli, decisions, and reward," *Neuron*, 36(2), 299–308.

Kleene, S.C. (1956), "Representation of events in nerve nets and finite automata," in *Automata Studies* eds. C.E. Shannon and J. McCarthy, Princeton, NJ: Princeton University Press, pp. 3–41.

Kopecz, K., and Schoner, G. (1995), "Saccadic motor planning by integrating visual information and pre-information on neural dynamics," *Biological Cybernetics*, 73, 49–60.

Usher, M., and McClelland, J.L. (2001), "The time course of perceptual choice: the leaky, competing accumulator model," *Psychological Review*, 108(3), 550–592.

Simen, P.A., and Cohen, J.D. (2009), "Explicit melioration by a neural diffusion model," *Brain Research*, 1299, 95–117.

Simen, P.A., Cohen, J.D., and Holmes, P. (2006), "Rapid decision threshold modulation by reward rate in a neural network," *Neural Networks*, 19, 1013–1026.

Gardiner, C.W., *Handbook of Stochastic Methods*, third ed., New York, NY: Springer-Verlag (2004).

Smith, P.L. (2010), "From Poisson shot noise to the integrated Ornstein-Uhlenbeck process: Neurally principled models of information accumulation in decision-making and response time," *Journal of Mathematical Psychology*, 54, 266–283.

Simen, P.A., and Polk, T.A. (2010), "A symbolic/subsymbolic interface protocol for cognitive modeling," *Logic Journal of the IGPL*, 18, 705–761.

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., and Cohen, J.D. (2006), "The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced choice tasks," *Psychological Review*, 113(4), 700–765.

Newell, A., *Unified Theories of Cognition*, Cambridge, MA: Harvard University Press (1990).

McMillen, T., and Holmes, P. (2006), "The dynamics of choice among multiple alternatives," *Journal of Mathematical Psychology*, 50, 30–57.

Simen, P.A., McMillen, T., and Behseta, S. (2010), "Hebbian learning for deciding optimally among many alternatives (almost)," in *Proceedings of the 2010 Cognitive Science Society conference*.

Hummel, J.E., and Holyoak, K.J. (1997), "Distributed representations of structure: a theory of analogical access and mapping," *Psychological Review*, 104(3), 427–466.

Ratcliff, R., and McKoon, G. (2008), "The diffusion decision model: theory and data for two-choice decision tasks," *Neural Computation*, 20, 873–922.

Russell, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall (1995).