*Review Article*

# Deep Blue Cannot Play Checkers: The Need for Generalized Intelligence for Mobile Robots

## Troy D. Kelley[1] and Lyle N. Long[2]

[1] *U.S. Army Research Laboratory, AMSRD-ARL-HR-SE, Aberdeen Proving Ground, MD 21005-5425, USA*
[2] *Department of Engineering and Mathematics, The Pennsylvania State University, University Park, PA 16802, USA*

Correspondence should be addressed to Lyle N. Long, lnl@psu.edu

Generalized intelligence is much more difficult than originally anticipated when Artificial Intelligence (AI) was first introduced in the early 1960s. Deep Blue, the chess playing supercomputer, was developed to defeat the top rated human chess player and successfully did so by defeating Gary Kasporov in 1997. However, Deep Blue only played chess; it did not play checkers, or any other games. Other examples of AI programs which learned and played games were successful at specific tasks, but generalizing the learned behavior to other domains was not attempted. So the question remains: Why is generalized intelligence so difficult? If complex tasks require a significant amount of development, time and task generalization is not easily accomplished, then a significant amount of effort is going to be required to develop an intelligent system. This approach will require a system of systems approach that uses many AI techniques: neural networks, fuzzy logic, and cognitive architectures.

## 1. Introduction

The problem of generalized intelligence has been plaguing the field of Artificial Intelligence (AI) since its inception in the early 1960s. Researchers in the area of AI had once hoped that a generalized intelligent system would be able to "grease a car or read Shakespeare; tell a joke or play office politics" [1]. Instead, AI researchers found that the learned rules and hard coded knowledge developed in order to solve a specific task was not likely to be transferable to other tasks. The International Business Machines (IBM) Corporation was eventually able to develop a computer program which defeated the most highly ranked chess player in the world [2]. However, IBM's computer only accomplished one task— playing chess. It did not play backgammon or checkers. It did not read Shakespeare or grease a car. The real problem for AI turned out to be the brittleness of behavior, or the lack of generalization across behaviors. In other words—how does playing chess help with reading Shakespeare? Or how does learning to tell a joke help with learning office politics? It could be that very little knowledge can be transferred from playing chess to reading Shakespeare. If this is the case

then how can generalized intelligence ever be realized? This paper will examine the different aspects of generalization and whether it can be performed successfully by future computer programs or robots.

Generalized intelligence is especially important for mobile robots, whether they are in the air, on the ground, in space, or in water. There is a need for these robots to be both intelligent and autonomous (and they may become conscious too [3]). Current mobile robots have very little intelligence or autonomy. Intelligence would allow the robots to perceive their environment, plan their activities, solve problems, and learn from experience [4]. Autonomy would allow them to survive in a real-world environment without external control. Intelligence and autonomy will require a complex system of systems approach that is highly inter-connected, much like the human brain and central nervous system. The sensor input data will need to be processed in a hierarchical manner from low-level features to complex objects. In addition, learning will be crucial. It will not be possible to program these systems for all possible circumstances that they will encounter. It will also not be physically possible to write all the needed software and logic rules. They

will need to be trained and nurtured as are human infants and children, and they need to learn from experience.

## 2. Symbolic and Subsymbolic Generalization

The chess playing computer program developed by IBM grew out of the symbolic tradition of AI. The symbolic tradition emphasized symbolic manipulations of information for problem solving, for example, the blocks world problem, the water jug problem, and so forth. A symbolic system uses localized representations of knowledge for problem solving (i.e., concepts are represented in one place). Mathematics and language are symbolic systems of knowledge representation (this assumption has been challenged by neural network researchers [5]). Symbolic systems of knowledge representation are in contrast to subsymbolic or distributed representations of knowledge. Subsymbolic representations of knowledge are not localized (i.e., concepts can be represented across a collection of nodes as weights). Neural networks and computer vision algorithms are subsymbolic representations of knowledge.

Subsymbolic systems were once hoped to be a solution to complex recognition tasks [5] and indeed they are capable of recognizing a wide variety of functions. Neural networks are capable of perceiving and classifying the noisy outside world given the correct topology and enough training examples. However, neural networks still have a problem with generalization. Within neural network research the generalization problem is defined as the over-fitting or overtraining problem. Neural networks can be trained to recognize training data; however, if the training is conducted with too many iterations, then the network will *only* be able to recognize the training data. Specifically, the network will not be able to generalize to data from outside the training set—it will be "over trained" or "over fit" to the original training set. And while recognition of previously trained data is an important component of an AI system, generalization is more important for a robust intelligence system. A neural network which has succumbed to over fitting will not be able to generalize outside the original training set. This can be especially problematic in real-world dynamic environments where the outside world is less predictable.

This issue is also a problem for the neural network developer since a network's architecture can be "under fit" as well as "over fit." In the "under fit" condition, the network will not be able to recognize training data without a sufficient number of training iterations. Moreover, the problem of over fitting a neural network is related to a number of structural factors in the network (i.e., network size, number of hidden layers, number of training examples) in addition to the point of learning convergence. These factors can make the autonomous selection of a cut-off point for training difficult to determine since the determination is related to many other architectural and data factors. The problem has become harder than expected within the neural network community [6] and this will continue to limit the subsymbolic generalization of traditional neural network architectures.

Once a network has been trained, it could be easily assumed that new training examples could be added to the original training set and this would overcome the generalization problem. However, this leads to the catastrophic forgetting problem [7]. Once a neural network has been trained on a specific set of training examples, training the net on new examples without including the older examples leads to the loss of the older examples. The network "forgets" the previously learned material. For example, one could easily train a neural network to recognize the alphabet, but once it is trained to do that, it would be difficult to then have it also learn to recognize numbers without human intervention. This greatly affects the generalization of the network since it becomes difficult to add new information to the network. Or, a neural network for a mobile robot that is trained to recognize doors would be hard to expand to include windows.

Newer neural network architectures [8], however, have overcome some of these problems and there are solutions to the problem catastrophic forgetting with traditional neural networks [9]. A newer promising architecture for overcoming problems of catastrophic forgetting is the Adaptive Resonance Theory (ART) [10]. ART was specifically developed to overcome problems of associated with catastrophic forgetting by adding additional functionality to a typical neural network. Specifically, ART uses a feedback mechanism between different layers of the network which allows the network to automatically switch between stable and flexible modes of operation. Additionally, ART has a competitive network, which, based on some criteria, allows nodes in the network to compete and select a winner in classification tasks. These additional functional mechanisms allow for online continual learning without the destruction of previous learned information. Additionally, they allow for learning based on a few number of instances (so called one-trial learning) which is one of the defining hallmarks of human learning.

Another type of relatively new neural networks is spiking neural networks [11]. These are biologically plausible time-dependent networks that are especially good for forming episodic memories and for sensor processing. These more closely model human neurons and synapses using nonlinear differential equations for membrane voltages and time-dependent Hebbian learning for synapses. These have been implemented to learn object recognition tasks. As in the human brain, one could also implement neurogenesis and synaptogenesis to allow the network to keep learning without forgetting the existing information.

It would appear then that newer types of subsymbolic architectures will be needed to allow for increased generalization over and above traditional neural networks. These newer architectures appear to be the only solutions to the problems of subsymbolic generalization (see Table 1).

## 3. Subsumptive Architectures

Subsumptive architectures [12] were developed as an answer to the brittleness and lack of generalization found in traditional symbolic AI systems. In a jab at symbolic systems, Brooks [13] titled his seminal work on subsumptive architectures, "Elephants do not play chess" implying that

TABLE 1: Symbolic and subsymbolic distinctions.

|  | Symbolic | Subsymbolic |
|---|---|---|
| Data | Discrete Symbolic | Distributed Metric |
| Strengths | Readable Logical | Fault tolerant Adaptive |
| Weakness | Frame problem | Catastrophic Forgetting |

elephants don't need symbolic manipulation. In a subsumptive architecture, explicit representations of the world or the problem space are intentionally avoided to allow for relatively simple generalizations across environments. Rules are used to represent a problem space, much like a symbolic architecture, but these rules tend to be extremely simple (i.e., move forward, turn left, move to light). These rules are embodied within agents and the rules compete for behavioral priority. For example, "turn left" would compete with "turn right," and would only execute if the threshold for the execution of "turn right" were, for some reason, lower than the threshold for the execution of "turn left." Subsumptive architectures do not represent the world as a model, because, as the subsumptive researchers explain, "the world is its own best model." The logic here is—why go about memorizing previous experiences and creating and updating a world model when one only has to look at the world to determine how to behave?

The answer to this question is that the world does not hold all the answers to all the questions and some questions are too abstract to represent with simple if-then statements. Learning and memory are very important to generalized intelligence. The problem with subsumptive architectures is that they work well for some simple behaviors, but more complex behavior is more difficult to represent. A subsumptive architecture would have difficulty playing a good game of chess. It might also produce mobile robots that get stuck in environments such cul-de-sacs. It is possible that a subsumptive architecture could play a game of chess, and perhaps get better at playing chess, but not reach a level of a grand master without formal symbolic representations. It is this very lack of symbolic complexity that leads to problems with subsumptive architectures. A reactive system might be suitable for lower level reactive behaviors of the type exhibited by an elephant, but what if the task domain required the symbolic manipulations necessary to play chess?

## 4. Cognitive Architectures

Cognitive architectures are well-known symbolic AI approaches that attempt to mimic human cognitive abilities via rule-based processing. Examples of these are Soar [14], ACT-R [15], and EPIC [16]. Some of these have been implemented on mobile robots [4, 17].

Some generalization can be accomplished using simple, rule-based, symbolic representations of knowledge. They can use general rule sets in order to encompass a wide variety

of structural mappings and thus allow for more robust decision making. For example, a door can be defined as an opening that leads into, or out of, a predefined space. Note that this definition says nothing about the specific properties of the door or even the specific properties of the space. Having such a broad, or generalized, definition of a "door" allows for a significant amount of generalization that is not specific to the properties of a door. For instance, using our previous definition, if a garage can be defined as a pre defined space, then any door leading into or out of the garage could be defined as also being a "door." So the definition applies to a garage door—even though a typical garage door has structural properties which are very different from a typical household door (i.e., a garage door is very large, opens horizontally, and typically does not include a doorknob while a household door is smaller, opens vertically, and usually includes a doorknob). The specification of an abstract or general rule can enable a system to perform symbolic generalizations across the underlying structural similarities of a problem space.

However, one problem with symbolic generalization techniques, especially those which generate a large number of possible actions, is the frame problem [18]. The frame problem can generally be thought of as having to represent every possibility symbolically. The discovery of the frame problem was one of the essential motivating factors in the development of the previously mentioned subsumptive architectures [12]. Interestingly, the frame problem goes away if development concentrates on task-specific behavior, which may be one reason why task-specific development has dominated AI research. It was an effort to ignore or overcome the frame problem. Task specific behavior, by very definition, is constrained, so developers did not have to worry about a wide variety of possible behaviors.

While the cognitive architectures are powerful and useful, they too have their limitations. A purely symbolic approach is difficult to use for processing detailed sensor data, for example. It would also be difficult to use it for sensitive motor output tasks. In addition, it is quite difficult to add learning to cognitive architectures, which is essential for future mobile robots. And unless the systems can develop their own rules, they are susceptible to the frame problem as well.

## 5. Hybrid Approach to Complex Cognition

Both symbolic and subsymbolic data and data processing have their advantages and disadvantages for mobile robots, and a hybrid approach using both are really required.

Human cognitive systems, and some intelligent animals, exhibit within their neurological functionality indications of *both* symbolic and subsymbolic representations of knowledge [19]. This is a *functional* distinction and not a cellular or neurological distinction. Specifically, the cerebral cortex and frontal lobes of the human cognitive system are capable of accomplishing symbolic manipulations of information while other parts of the system appear to operate—functionally— in a more distributed manner. However, the cerebral cortex and frontal lobes are probably neurologically distributed
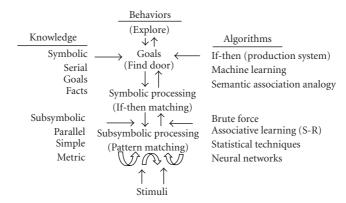
FIGURE 1: SS-RICS notional data flow with the symbolic and subsymbolic distinctions [3].

systems—at least in terms of memory components [20], but in this paper, we are making a distinction between the functional aspects not the structural aspects. We are making this distinction between different types of knowledge representation (i.e., symbolic versus subsymbolic) because the type of knowledge representation used within an intelligent system will affect the ways in which the information can be manipulated.

There are several examples of hybrid systems: SS-RICS [17], CRS [4], and CLARION [21].

SS-RICS is largely based on the Adaptive Control of Thought-Rational (ACT-R) (see Figure 1). It implements a production system architecture similar to ACT-R with decay algorithms that affect memories. Like ACT-R, it is production system architecture at the highest level, which we are using to mimic the functionality of human working memory. SS-RICS also includes a subsymbolic system at the lower levels, which mimic iconic short-term memory and perception. The production system within SS-RICS is composed of rules and goals. SS-RICS can also access "concepts" which are long term facts; similar to declarative memories within ACT-R, but within SS-RICS they are considered long term memories (i.e., memories which do not decay). SS-RICS can also generate productions automatically in order to generalize [17]. SS-RICS has two types of processes for the symbolic generation of new productions, top-down learning and bottom-up learning. It is envisioned that SS-RICS will have other types of symbolic generation mechanisms primarily because there seem to be a number of other techniques already available [22].

CRS uses Soar combined with subsymbolic processing (e.g., computer vision systems). In this case, Soar is coupled to Java software for input and output processing. The sensor inputs and motor control outputs are controlled in Java while the symbolic data is stored in Soar. CRS has been used on both wheeled and legged robots [4]. It has also been used with several types of sensor input systems (vision, sonar, GPS, compass, touch, etc.). The approach has been very effective, and it is fairly easy to add additional sensors, rules, or output devices.

The Connectionist Learning with Adaptive Rule Induction ON-line, or CLARION cognitive architecture is designed to capture the interaction between subsymbolic and symbolic processes, or as the CLARION developers say, the distinction between implicit and explicit knowledge. CLARION uses Q learning, or reinforcement learning, at subsymbolic levels. Additionally, the architecture uses rule extraction algorithms at the symbolic level to develop links between subsymbolic and symbolic information.

Generalized symbolic representations work well with real-world data as compared to subsymbolic and statistical representations. In a recent exercise at the National Institutes of Standards and Technology (NIST) using SS-RICS (February, 2009) we found that a symbolic representation of the intersections in a maze was just as useful as neural network representations or statistical representations. We essentially used "primitives" to define a small core set of examples which were predefined by the programmer. The use of primitives seems to work well as an overall strategy for symbolic generalization. This is essentially the same strategy as the use of "scripts" [23] or "frames" [24] where abstract symbolic representations are used to represent a variety of problem situations (e.g., restaurants). This approach seems to work well for symbolic representations of knowledge.

## 6. Human Generalization

In order for humans to successfully accomplish a generalized learning procedure, a number of complex operations or components are required. Each one of the components within the generalization process can be difficult for humans to accomplish effectively. Researchers [25] have identified four major components of the analogy or generalization processes.

(1) The retrieval or selection of a plausibly useful source analog.

(2) Mapping.

(3) Analogical inference or transfer.

(4) Subsequent learning.

Gick and Holyoak [26, 27] found that in the absence of clear relationships between two problem spaces useful analogies are often not discovered by problem solvers. Additionally, Holyoak and Koh [25] found that subjects are

frequently fixated on the salient surface regularities of a problem space and not the underlying structural features, making the selection of a useful source analog even more problematic. Moreover, mapping can be a difficult process as well, because it is not only the selection of a source analog but also selecting which aspects of the source are important [28]. Additionally, Novick [29] found instances of negative transfer—where misleading surface similarities were used to create the analogy. So it would appear from the human behavioral data that human generalization is not easily accomplished by human problem solvers.

To complicate matters, it would also appear that mastering a task to the level of an expert is also quite difficult. Ericsson [30] has found that in order to develop expertise in a complex task, a human subject needs to have a considerable amount of time doing the task. This amount of time, according to Ericsson, is roughly estimated to be 10,000 hours (roughly 5 years of full-time effort). Even human experts who were once considered to be child prodigies, like Wolfgang Mozart or Bobby Fisher, still required 10,000 hours before they were able to perform at a level that was considered a "master" level. Additionally, not only does task-specific behavior require a large amount of experience but it also requires "deliberate practice" in order for an expert to fully master a complex task (like playing chess). Ericsson has shown that human masters in a specific task require a large amount of practice before they become proficient at the task and that the practice has to be efficient and useful for a full development of skills. The human data from Ericsson's studies seem to be concurrent with computer software data: that is, in order to develop a system with expertise in one area a significant amount of development time is required. This would imply that task-specific behavior requires a large amount of learning or development time before an autonomous system would be capable of mastering a task.

If generalization is difficult for human subjects and task-specific behavior is time consuming for human subjects then one must only assume that generalization for robotics and autonomous systems will also be difficult. AI researchers have found that the development of complex task-specific behavior requires an enormous amount of hand tuning in order to achieve the desired results and this would seem to apply to human learning as well. Generalization has proved difficult for human subjects and this seems to also be the case for AI systems as well. The two problems are related. The more task-specific behavior that is developed, the more likely a system can generalize to a new environment. Thrun [31] has noted that learning becomes easier when embedded in a life-long context. So, it could be then that the development of task-specific behaviors will help with generalization—but only if the bulk of the behaviors can be applied to a new situation.

## 7. Hybrid Intelligent Systems for Mobile Robots

Until we can reverse engineer human or animal brains, we will need to use a clever assortment of algorithms in order to build intelligent, autonomous, and possibly conscious mobile robots. We will need to take a system of systems

approach. The various sensor inputs (vision, touch, smell, sound, etc.) will each need to be highly refined systems, and these will most likely be subsymbolic systems as they are in humans. We will also need symbolic systems that use fuzzy logic, rule-based approaches, and other AI techniques. And we will also need motor control output systems. There are roughly 600 muscles in the human body, each controlled by the central nervous system.

It is also important to recognize that whether an approach is considered subsymbolic or symbolic really depends on the granularity of the view taken. In the human brain, all the processes are cell or neuron-based. So at that level everything is subsymbolic. Groups of neurons working together, however, can perform tasks that appear symbolic. For example, in the human vision system there are face recognition subsystems. The human brain also has place cell subsystems, and possibly "grandmother cells" subsystems. Another example is the use of fuzzy logic. It has been shown that neural networks can be replaced by fuzzy logic systems, while the neural network would appear to be a subsymbolic system the ultimate function of the system could be replaced by a symbolic system. Since, at the current time, we cannot model all the roughly $10^{11}$ neurons or $10^{14}$ synapses (or the wiring diagram that connects them all), we will need to model some of these systems using symbolic approaches. Intelligence is defined as: "a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly, and learn from experience" [32].

Current cognitive architectures do have the ability to perform some of these tasks, but in order to accomplish many of these the robot will need the abilities discussed earlier regarding analogies and inference. Autonomy is defined as: The ability to operate "in the real-world environment without any form of external control for extended periods of time" [33].

Current mobile robots are usually designed for very specific environments, and even indoor and outdoor robots are designed very differently. The ultimate goal should be to build robots that can adapt to new environments or can be taught and nurtured by humans for the various environments.

It will not be possible to completely wire or program these robots, however. They will need to learn on their own, through traditional machine learning approaches, traditional neural network learning, or neuro/synaptogenesis. Learning will be required at both the symbolic and subsymbolic levels. For the sensor input or motor-control output systems, we will most likely need subsymbolic learning, while at the symbolic level we will need more traditional machine learning approaches. The human brain is possibly the most complex system in the known universe, we must also remember that it takes roughly 20 years to train this system, and roughly 5 years to become an expert on some topic. For robots to reach human levels of intelligence and autonomy the training and learning will need to be extensive. Once it has been trained though, it can then be easily replicated!

We have done some initial work with conceptual generalization and we have found that instance-based recognition of

objects is relatively easy, but that generalized recognition of objects is much more difficult. For example, during robotic navigation, a robot needs to be able to recognize doors as landmarks. Using simple image correlation algorithms (i.e., template matching), we were able to train SS-RICS to recognize specific instances of doors (i.e., the front door, the back door, Sue's door). However, the image correlation algorithm was easily fooled by lighting variations or occlusion. Additionally, the algorithm would sometimes not work with objects that had not been seen before, especially if the new image was significantly different from the template. In addition to using image correlation, we also have used other algorithms for shape definitions. These algorithms extracted the shape of the object being viewed. For instance, a door shape is typically rectangular. However, these algorithms can be problematic if the door is partially occluded. Also, the shape algorithm worked best with a extremely clean data, which is not typical in real world conditions.

We have hypothesized that developing generalized recognition emerges from previous exposures to numerous instance based recognitions. The essential component to this hypothesis is a prior expectation of the input data. In order to account for changes across time due to lighting variations an algorithm is being developed that checks to see if the image being viewed is somehow different from what is expected (correlation). This obviously requires an expectation metric for each object being viewed. If the input data is somehow lower than what is expected, the algorithm first needs to assume that what is being viewed is the previously viewed object but it has somehow changed, and not a new object. This assumption is not always correct, but for the initial implementation of the algorithm, this was the default assumption. Next, the algorithm checks to see what has remained the same across the two images. The features that have remained the same across the two images would represent the essential features of the object. The essential features are what need to be learned to allow for generalization.

## 8. Conclusions

We have discussed the idea that both symbolic and subsymbolic generalization are difficult for computational systems. We have also discussed the research that shows that generalization is difficult for people as well as computer systems. However, while subsymbolic generalization is difficult for computer systems it is relatively *easy* for people. Human subjects can learn from one example and do not seem to suffer from catastrophic forgetting in the same way that traditional subsymbolic systems do. People can be exposed to just one example of a stimulus and be able to incorporate this new example into a generalized understanding of the stimulus (a.k.a. one-trial learning). One would assume then that the mechanisms of subsymbolic generalization used by humans must somehow be different from the traditional generalizations in subsymbolic systems. However, perhaps the newer subsymbolic algorithms [8–10, 34] will lead to more robust subsymbolic generalizations.

Task specific behaviors, which have been extensively developed by AI systems over the last 40 years, require a large amount of development time and programmer expertise. The continuation of task-specific development must be approached with longer term goals of modularity and reuse in order to facilitate broader generalization. A long development time for task-specific behavior is also reflected in the human data associated with acquisition of expert performance [28]. If task generalization becomes easier when learning is done within the context of extensive previous knowledge [29], then generalization can only proceed once an extensive amount of task-specific behavior has already been developed. This would argue for the continuation of task-specific approaches, but with the need for generalization to occur following the development of task-specific behaviors. This seems to have been ignored by much of the AI research. If we assume the need for an extensive knowledge base, and this is represented symbolically—primitives, scripts, or frames can be used to generalize across problem spaces. But this only applies to symbolic representations of knowledge not subsymbolic representations of knowledge.

Subsumptive architectures are capable of task generalization across simple tasks. However, the very mechanisms that make subsumptive architectures powerful generalization systems within a simple task domain are the same mechanisms that exclude them from being able to generalize across more complex task domains (i.e., lack of a world model). This would argue for a combination of approaches, with a subsumptive architecture being able to generalize over simple reactive tasks coupled with a symbolic system for generalization across more complex tasks. Some researchers have called for the combinations of Bayesian/probabilistic and connectionist approaches [34]. This would facilitate further understanding of the applicability of subsymbolic knowledge structures. In truth, there is probably a continuum of approaches that span the range between symbolic and subsymbolic approaches to include statistical, Bayesian, and chaotic knowledge structures.

Perhaps another approach to accomplishing a generalized intelligent system capable of performing a wide variety of tasks would be something similar to the DARPA Grand Challenge [35] except instead of researching mobility across an open road, the task would be researching generalization across tasks. This would help to emphasize the architectural constraints associated with generalization within each architecture and would force researchers to address how their respective architectures would generalize to different environments. Perhaps the "winning" architecture could then be accepted by researchers and then allowed to learn other tasks on the way to a generalized learning system.

The prospects for a generalized intelligence system are daunting. We have argued in previous papers [3, 36] that complex cognition will require a complex approach Both symbolic and subsymbolic systems appear to have some limitations with regard to generalization, however, newer subsymbolic are capable of addressing some of the limitations associated with subsymbolic learning. Symbolic systems show some promise with generalization given enough prior information and the use of frames or scripts. The careful

combination of symbolic systems with newer subsymbolic structures, along with an extensive experiential knowledge base, all appear to be necessary to solving the generalization challenge. Only then will intelligent and autonomous mobile robots be possible.

# References

[1] B. Darrach, "Meet Shaky: the first electronic person," *Life Magazine*, vol. 69, no. 21, pp. 58B–68B, 1970.

[2] D. Goodman and R. Keene, *Man Versus Machine, Kasparov Versus Deep Blue*, H3 Publications, Cambridge, Mass, USA, 1997.

[3] L. N. Long and T. D. Kelley, "Review of consciousness and the possibility of conscious robots," *Journal of Aerospace Computing, Information and Communication*, vol. 7, no. 2, pp. 68–84, 2010.

[4] S. D. Hanford, O. Janrathitikarn, and L. N. Long, "Control of mobile robots using the soar cognitive architecture," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 2, pp. 69–71, 2009.

[5] D. E. Rumelhart and J. L. McClelland, "On learning the past tenses of English verbs," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, J. L. McClelland and D. E. Rumelhart, Eds., vol. 11, pp. 216–270, MIT Press, Cambridge, Mass, USA, 1986.

[6] S. Lawrence, C. L. E. E. Giles, and A. H. C. Tsoi, "Lessons in neural network training: overfitting may be harder than expected," in *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pp. 540–545, Menlo Park, Calif, USA, August 1997.

[7] M. McCloskey and N. Cohen, "Catastrophic interference in connectionist networks: the sequential learning problem," in *The Psychology of Learning and Motivation*, G. H. Bower, Ed., vol. 24, pp. 109–164, Academic Press, New York, NY, USA, 1989.

[8] S. E. Fahlman and C. Lebiere, "The cascade-correlation architecture," in *Advances in Neural Information Processing Structures*, D. S. Touretsky, Ed., vol. 2, pp. 524–532, Morgan Kaufmann, San Mateo, Calif, USA, 1990.

[9] A. V. Robins and S. J. R. McCallum, "A robust method for distinguishing between learned and spurious attractors," *Neural Networks*, vol. 17, no. 3, pp. 313–326, 2004.

[10] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory (ART)," *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2002.

[11] W. Maass and C. M. Bishop, *Pulsed Neural Networks*, MIT Press, Cambridge, Mass, USA, 1999.

[12] R. Pfeifer and C. Scheier, *Understanding Intelligence*, MIT Press, Cambridge, Mass, USA, 2000.

[13] R. A. Brooks, "Elephants don't play chess," *Robotics and Autonomous Systems*, vol. 6, no. 1-2, pp. 3–15, 1990.

[14] A. Newell, *Soar: A Cognitive Architecture in Perspective*, Harvard University Press, Cambridge, Mass, USA, 1990.

[15] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*, Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1998.

[16] D. E. Meyer and D. E. Kieras, "A computational theory of executive cognitive processes and multiple-task performance: part 1. Basic mechanisms," *Psychological Review*, vol. 104, no. 1, pp. 3–65, 1997.

[17] T. D. Kelley, E. Avery, L. N. Long, and E. Dimperio, "A hybrid symbolic and sub-symbolic intelligent system for mobile robots," in *InfoTech@Aerospace Conference*, Seattle, Wash, USA, 2009, AIAA 2009-1976.

[18] J. McCarthy and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," *Machine Intelligence*, vol. 4, pp. 463–502, 1969.

[19] T. D. Kelley, "Symbolic and sub-symbolic representations in computational models of human cognition: what can be learned from biology?" *Theory and Psychology*, vol. 13, no. 6, pp. 847–860, 2003.

[20] P. Dean, "Recapitulation of a theme by Lashley? Comment on Wood's simulated lesion experiment," *Psychological Review*, vol. 87, no. 5, pp. 470–473, 1980.

[21] R. Sun and T. Peterson, "Hybrid learning incorporating neural and symbolic processes," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 727–732, Anchorage, Alaska, USA, June 1998.

[22] T. A. Mitchell, *Machine Learning*, WCB/McGraw Hill, Boston, Mass, USA, 1997.

[23] R. C. Schank, "The structure of episodes in memory," in *Representation and Understanding: Studies in Cognitive Science*, D. Bobrow and A. Collins, Eds., Academic Press, New York, NY, USA, 1975.

[24] M. Minsky, "A framework for representing knowledge," in *Mind Design: Philosophy, Psychology, Artificial Intelligence*, J. Haugeland, Ed., MIT Press, Cambridge, Mass, USA, 1975.

[25] K. J. Holyoak and K. Koh, "Surface and structural similarity in analogical transfer," *Memory and Cognition*, vol. 15, no. 4, pp. 332–340, 1987.

[26] M. L. Gick and K. J. Holyoak, "Analogical problem solving," *Cognitive Psychology*, vol. 12, pp. 306–355, 1980.

[27] M. L. Gick and K. J. Holyoak, "Schema induction and analogical transfer," *Cognitive Psychology*, vol. 15, no. 1, pp. 1–38, 1983.

[28] K. J. Holyoak and P. Thagard, "Analogical mapping by constraint satisfaction," *Cognitive Science*, vol. 13, no. 3, pp. 295–355, 1989.

[29] L. R. Novick, "Analogical transfer, problem similarity, and expertise," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 14, no. 3, pp. 510–520, 1988.

[30] K. A. Ericsson, "The influence of experience and deliberate practice on the development of superior expert performance," in *Cambridge Handbook of Expertise and Expert Performance*, K. A. Ericsson, N. Charness, P. Feltovich, and R. R. Hoffman, Eds., Cambridge University Press, Cambridge, UK, 2006.

[31] S. Thrun, "Lifelong learning algorithms," in *Learning to Learn*, S. Thrun and L. Pratt, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 1998.

[32] L. S. Gottfredson, "Mainstream science on intelligence: an editorial with 52 signatories, history, and bibliography," *Intelligence*, vol. 24, no. 1, pp. 13–23, 1997.

[33] G. A. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control*, MIT Press, Cambridge, Mass, USA, 2005.

[34] A. Gupta and L. N. Long, "Hebbian learning with winner take all for spiking neural networks," in *Proceedings of International Joint Conference on Neural Networks (IJCNN '09)*, pp. 1054–1060, Atlanta, Ga, USA, June 2009.

[35] J. L. McClelland, "The place of modeling in cognitive science," *Topics in Cognitive Science*, vol. 1, no. 1, pp. 11–38, 2009.

[36] T. D. Kelley, "Simulating intelligent behavior requires a complex approach," in *Proceedings of the AAAI Spring Symposium on Between a Rock and a Hard Place: Cognitive Science Principles Meet AI-Hard Problems*, C. Lebiere and R. Wray, Eds., pp. 70–73, AAAI press, Stanford, Calif, USA, March 2006.