A Minimal Architecture for General Cognition

Michael S. Gashler
Department of Computer Science
and Computer Engineering
University of Arkansas
Fayetteville, Arkansas 72701, USA
mgashler@uark.edu

Zachariah Kindle
Department of Computer Science
and Computer Engineering
University of Arkansas
Fayetteville, Arkansas 72701, USA
zkindle@email.uark.edu

Michael R. Smith
Department of Computer Science
Brigham Young University
Provo, Utah 84602, USA
msmith@axon.cs.byu.edu

***Abstract*—A minimalistic cognitive architecture called MANIC is presented. The MANIC architecture requires only three function approximating models, and one state machine. Even with so few major components, it is theoretically sufficient to achieve functional equivalence with all other cognitive architectures, and can be practically trained. Instead of seeking to trasfer architectural inspiration from biology into artificial intelligence, MANIC seeks to minimize novelty and follow the most well-established constructs that have evolved within various subfields of data science. From this perspective, MANIC offers an alternate approach to a long-standing objective of artificial intelligence. This paper provides a theoretical analysis of the MANIC architecture.**

I. Introduction

When artificial intelligence was first discussed in the 1950's as a research project at Dartmouth [1], many of the early pioneers in AI optimistically looked forward to creating intelligent agents. In 1965 Herbert Simon predicted that "machine will be capable, within twenty years, of doing any work man can do" [2]. In 1967, Marvin Minsky agreed, writing "within a generation …the problem of creating 'artificial intelligence' will be substantially solved" [3]. Since that time, artificial general intelligence has not yet been achieved as predicted. Artificial intelligence has been successfully applied in several specific applications such as natural language processing and computer vision. However, these domains are now subfields and sometimes even divided into sub-subfields where there is often little communication [4].

Identifying the minimal architecture necessary for a particular task is an important step for focusing subsequent efforts to implement practical solutions. For example, the Turing machine, which defined the minimal architecture necessary for general computation, served as the basis for subsequent implementations of general purpose computers. In this paper, we present a Minimal Architecture Necessary for Intelligent Cognition (*MANIC*). We show that MANIC simultaneously achieves theoretical sufficiency for general cognition while being practical to train. Additionally, we identify a few interesting parallels that may be analogous with human cognition.

In biological evolution, the fittest creatures are more likely to survive. Analogously, in computational science, the most effective algorithms are more likely to be used. It is natural to suppose, therefore, that the constructs that evolve within the computational sciences might begin, to some extent, to mirror those that have already evolved in biological systems. Thus, inspiration for an effective cognitive architure need not necessarily originate from biology, but may also come from the structures that self-organize within the fields of machine learning and artificial intelligence. Our ideas are not rooted in the observations of human intelligence, but in seeking a simplistic architecture that could plausibly explain high-level human-like cognitive functionality.

In this context, we propose that high-level cognitive functionality can be explained with three primary capabilities: 1- environmental perception, 2- planning, and 3- sentience. For this discussion, we define **Environmental perception** as an understanding of the agent's environment and situation. An agent perceives its environment if it models the environment to a sufficient extent that it can accurately anticipate the consequences of candidate actions. We define **Planning** to refer to an ability to choose actions that lead to desirable outcomes, particularly under conditions that differ from those previously encountered. We define **Sentience** to refer to an awareness of feelings that summarize an agent's condition, coupled with a desire to respond to them. We carefully show that MANIC satisfies criteria 1 and 2. Criterion 3 contains aspects that are not yet well-defined, but we also propose a plausible theory for sentience to which MANIC can achieve functional equivalence.

Figure 1 shows a subjective plot of several representative cognitive challenges. This plot attempts to rank these challenges according to our first two capabilites: perception and planning. Due to the subjective nature of these evaluations exact positions on this chart may be debated, but it is significant to note that the challenges typically considered to be most representative of human cognition are those that simultaneously require capabilities in both perception and planning. Problems requiring only one of these two abilities have been largely solved in the respective sub-discplines of machine learning and artificial intelligence, in some cases even exceeding human capabilities. It follows that human-like cognition requires an integration of the recent advances in both machine learning and artificial intelligence. MANIC seeks to identify a natural integration of components developed in these respective fields.

This document is laid out as follows: Section II describes the MANIC cognitive architecture. Section III shows that MANIC is theoretically sufficient to accomplish general cognition. Section IV describes how the MANIC cognitive architecture can be practically trained using existing methods. Section V discusses a plausible theory for sentience, and describes how the MANIC architecture can achieve functional

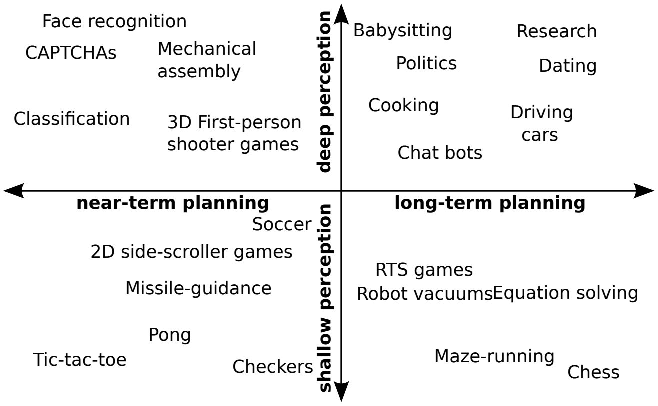

Fig. 1. A subjective plot of several cognitive challenges. Advances in machine learning, especially with deep artifical neural networks, have solved many problems that require deep perception (top-left quadrant). Advances in artificial intelligence have solved many problems that require long-term planning (bottom-right quadrant). The unsolved challenges (mostly in the top-right quadrant) require a combination of both deep perception and long-term planning. Hence, the necessity of cognitive architectures, which combine advances in both sub-disciplines to address problems that require a combination of cognitive abilities.

equivalence with it. Finally, Section VI concludes by summarizing the contributions of this paper.

II. Architecture

An implementation of MANIC can be downloaded from https://github.com/mikegashler/manic.

A cognitive architecture describes a type of software agent. It operates in a world that may be either physical or virtual. It observes its world through a set of percepts, and operates on its world through a set of actions. Consistent with other simple artificial architectures, MANIC queries its percepts at regular time intervals, $t = 0, 1, 2, \ldots$, to receive corresponding vectors of observed values, $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$. It chooses action vectors to perform at each time, $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots$. For our discussion here, MANIC assumes that its percepts are implemented with a camera (or renderer in the case of a virtual world), such that each $\mathbf{x}_t$ is a visual digital image. We use vision because it is well understood in the context of demonstrating cognitive abilities. Other percepts, such as a microphone, could also be used to augment the agent's observations.

At a high level, the MANIC divides into two systems, which we call the *learning system* and the *decision-making system*. The learning system draws its architecture from constructs that are studied predominantly in the field of machine learning, and the decision-making system draws its architecture from the constructs that are studied predominantly in the partly overlapping field of artificial intelligence. The agent's percepts feed into the learning system, providing the source of data necessary for learning, and the agent's actions derive from the decision-making system, which determines the action vectors for the agent to perform. A diagram of the MANIC architecture is given in Figure 2.

A. Learning system

The purpose of the learning system is to learn from past experiences. A large portion of the research effort in the field of machine learning thus far has focused on static systems. These are systems where the outputs (or labels) depend only on the current inputs (or features) and some component of unobservable noise. With a software agent, however, it is necessary for the agent to model its world as a dynamical system. (A dynamical system refers to one that changes over time, and should not be confused with a dynamic system, which is a different concept.)

Although many approaches exist for modeling dynamical systems in machine learning, nearly all of them either explicitly or implicitly divide the problem into three constituent components: a *transition model*, f, a *belief vector*, $\mathbf{v}_t$, and an *observation model*, g. The belief vector is an internal representation of the state of the system at any given time. In the context of a cognitive architecture, we refer to it as "belief" rather than "state" to emphasize that it is an intrinsic representation that is unlikely to completely represent the dynamical system, which in this case is the agent's entire world, that it attempts to model. The transition model maps from current beliefs to the beliefs in the next time step, $\mathbf{v}_{t+1} = f(\mathbf{v}_t)$. It is implemented using some function approximating regression model, such as a feed-forward multilayer perceptron. The observation model is a bi-directional mapping between anticipated beliefs and anticipated observations, $\hat{\mathbf{x}}_t = g(\mathbf{v}_t)$, and $\mathbf{v}_t = g^+(\mathbf{x}_t)$, where g^+ approximates the inverse of g.

When trained, this learning system enables the agent to anticipate observations into the arbitrarily distant future (with decaying accuracy) by beginning with the current beliefs, $\mathbf{v}_t$, then repeatedly applying the transition function to estimate the beliefs at some future time step, $\mathbf{v}_{t+i}$. The future beliefs may be passed through the observation model to anticipate what the agent expects to observe at that time step. Because the agent is unlikely to ever successfully model its complex world with perfect precision, the anticipated observations may differ somewhat from the actual observations that occur when the future time step arrives. This difference provides a useful error signal, $e = \mathbf{x}_t - \hat{\mathbf{x}}_t$, which can be utilized to refine the learning system over time. In other words, the agent knows its learning system is well trained when the error signal converges toward a steady stream of values close to zero.

This learning system represents the common intersection among a wide diversity of models for dynamical systems. For example, an Elman network [5] is a well-established recurrent neural network suitable for modeling dynamical systems. It is typically considered as a single network, but can be easily segmented into transition and observation components, and its internal activations may be termed a belief vector. A NARMAX model, which is commonly used in system identification more explicitly segments into these three components [6]. A Jordan network is a recurrent neural network that lacks an observation model component, but it may still be considered to be a degenerate case of an Elman network that uses the identity function for its observation component. Many other approaches, including the extended Kalman filter [7] and LSTM networks [8] naturally fit within this learning system design.

1) Transition model: Although the learning system as a whole is a recurrent model, the transition model may be implemented as a simple feed-forward model, because it only needs to predict $\mathbf{v}_{t+1}$ from $\langle \mathbf{v}_t, \mathbf{u}_t \rangle$.

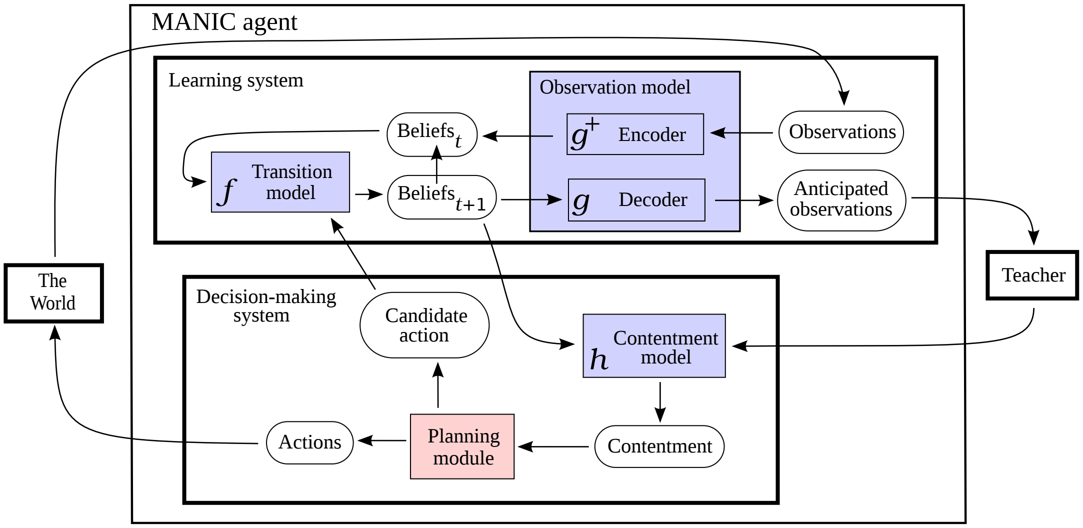

Fig. 2. A high-level diagram of MANIC, which consists of just 3 function approximating models (blue rectangles), one of which is bi-directional, and 1 state machine (red rectangle). Vectors are shown in rounded bubbles. At the highest level, MANIC divides its artificial brain in accordance with functional divisions that have evolved between the sub-disciplines of machine learning and artificial intelligence.

2) Observation model: The purpose of the observation model is to map between beliefs and observations. It is a bi-directional mapping between states and observations. Of all the components in MANIC, the observation model may arguably be the most critical for the overall success of the agent in achieving proficiency in general settings. If g and g^+ are well-implemented, then the agent's internal representation of beliefs will reflect a rich understanding of its world. This is one aspect of cognition where humans have traditionally excelled over machines, as evidenced by our innate ability to recognize and recall images so proficiently [9]. By contrast, machines have long been able to navigate complex decision chains with greater effectiveness than humans. For example, machines are unbeatable at checkers, and they can consistently trounce most humans at chess. Even decision-making processes that only utilize short-term planning may appear to exhibit much intelligence if they are based on a rich understanding of the situation.

Since about 2006, a relatively new research community has formed within the field of artificial neural networks to study "deep learning" architectures [10], [11]. These deep learning architectures have demonstrated significant advances in ability for mapping graphical images to hierarchical feature representations. For example, in 2009, Lee and others showed that deep networks decompose images into constituent parts, much as humans understand the same images [12]. In 2012, Krizhevsky demonstrated unprecidented accuracy at visual recognition tasks using a deep learning architecture trained on the ImageNet dataset [13], and many other developments in this community have eclipsed other approaches for image recognition. Since MANIC assumes that observations are given in the form of digital images, and the task of the observation model is to map from these observations to meaningful internal representations, the developments of this community are ideally suited to provide the best implementation for the observation model. The encoding component of the observation model, g^+, may be implemented as a deep convolutional neural network, which is known to be particularly effective for encoding images [14], [15], [16]. The decoding component, g, may be implemented as a classic fully-connected deep neural network. Instead of predicting an entire observation image, it models the image as a continuous function. That is, it predicts all the color channel values for only a single pixel, but additional inputs are added to specify the pixel of interest. Such an architecture can be shown to implement the decoding counterpart of a deep convolutional neural network.

If an absolutely minimal architecture is desired, the encoder, g^+, may be omitted. Rather than calculating beliefs from the current observations, beliefs may be refined to make the anticipated observations match the actual observations. This inference-based approach has the advantage of remembering believed state even when it is not visible in the current observations. Hence, the encoder is only used in the rare cases when the agent is activated in a new context, but repeated applications of inference with the decoder would likely accomplish the same objective.

B. *Decision-making system*

The decision making system produces plans to accomplish tasks that are anticipated to maximize the system's contentment. Some clear patterns have also emerged among the many decision-making systems developed for artificial intelligence. Nearly all of them divide the problem into some form of model for estimating the utility of various possible states, and some method for planning actions that are expected to lead to desirable states. Let $h(\mathbf{v}_t)$ be the utility that the agent believes to be associated with being in state $\mathbf{v}_t$. Let $\mathbf{P}$ be a pool of candidate plans for maximizing h that the agent considers, where each $\mathbf{p}_i \in \mathbf{P}$ is a sequence of action vectors, $\mathbf{p}_i = \langle \mathbf{u}_t, \mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \cdots \rangle$. At each time step, the agent performs some amount of refinement to its pool of plans, selects the one that yields the biggest expected utility, and chooses to perform the first action in that plan.

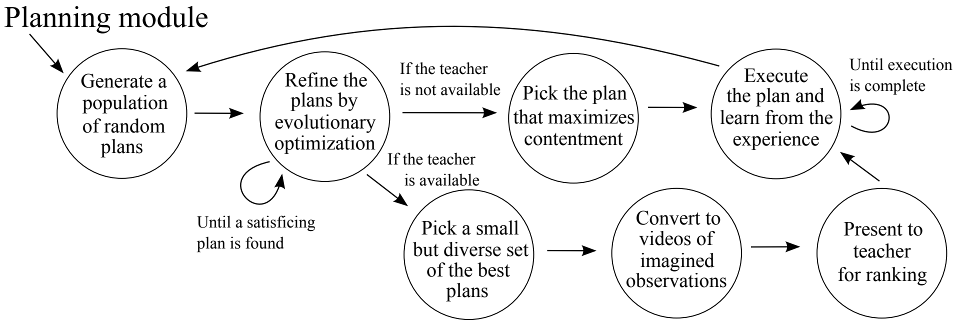
Planning module
Generate a population of random plans
Refine the plans by evolutionary optimization
Until a satisficing plan is found
If the teacher is not available
If the teacher is available
Pick the plan that maximizes contentment
Execute the plan and learn from the experience
Until execution is complete
Pick a small but diverse set of the best plans
Convert to videos of imagined observations
Present to teacher for ranking

Fig. 3. The planning module is state machine that makes plans by using the 3 function-approximating models to anticipate the consequences and desirability of candidate action sequences.

At this high level, the model is designed to be sufficiently general to encapsulate most decision-making processes. For example, those that maintain only one plan, or look only one time-step ahead, may be considered to be degenerate cases of this model with a small pool or short plans. Processes that do not refine their plans after each step may implement the regular refinement as an empty operation. Implementation details of this nature are more properly defined in the lower-level components.

1) Contentment model: Because MANIC is designed to be a long-living agent, we think of utility as being maintained over time, rather only being maximized in carrying out a single plan. Thus, we refer to h as the *contentment model*. We assume that any tasks the agent is intended to accomplish are achieved through the agent's constant efforts to preserve a state of homeostasis. For example, a robotic vacuum would be most "content" when the floor has been recently cleaned. In order to maintain this condition of contentment, however, it may need to temporarily stop cleaning in order empty its bag of dust, recharge its batteries, or avoid interfering with the owner. The contentment model, h, is trained to capture all of the factors that motivate the agent.

The contentment model is the only component of MANIC that benefits from a human teacher. Because the models in the learning system learn from unlabeled observations, direction from a teacher can be focused toward this one model. Competitive approaches can also be used to train the contentment model without a human teacher, as described in Section IV.

2) Planning module: The planning module is a simple state machine that utilizes the three models to anticipate the outcome of possible plans, and evaluate their utility. A flow chart for this module is given in Figure 3. Much work has been done in the field of artificial intelligence to develop systems that guarantee convergence toward optimal plans. Unfortunately, these systems typically offer limited scalability with respect to dimensionality [17]. Early attempts at artificial intelligence found that it is much easier to exceed human capabilities for planning than it is to exceed human capabilities for recognition [9]. This implies that humans utilize rich internal representations of their beliefs to compensate for their relative inability to plan very far ahead. To approach human cognition, therefore, it is necessary to prioritize the scalability of the belief-space over the optimality of the planning. A good compromise with many desirable properties is found in genetic algorithms that optimize by simulated evolution. These methods maintain a population of candidate plans, can benefit from prior planning, are easily parallelized, and scale very well into high-dimensional belief spaces.

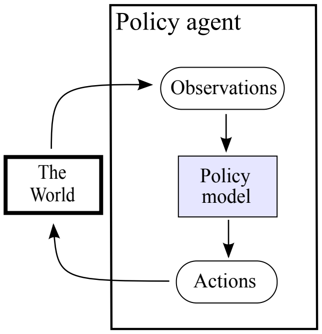

Fig. 4. A simple architecture that is not sufficient for general cognition.

III. Sufficiency

Occam's razor suggests that additional complexity should not be added to a model without necessity. In the case of cognitive architectures, additional complexity should probably give the architecture additional cognitive capabilities, or else its necessity should be challenged. In this section, we evaluate MANIC against Occam's razor, and contrast it with two even simpler models to highlight its desirable properties.

A diagram of a simple *policy agent* is given in Figure 4. This agent uses a function approximating model to map from the current observations to actions. (It differs from a reflex agent [18] in that its model is not hard coded for a particular problem, but may be trained to approximate solutions to new problems that it encounters.) The capabilities of this architecture are maximized when it is implemented with a model that is known to be capable of approximating arbitrary functions, such as a feedforward artificial multilayer perceptron with at least one hidden layer [19], [20]. However, no matter how powerful its one model may be, there are many problems this architecture cannot solve that more capable architectures can solve. For example, if observations are not sufficient to uniqely identify state, an architecture with memory could do better. Therefore, we say that this policy agent is not sufficient to implement general cognition.

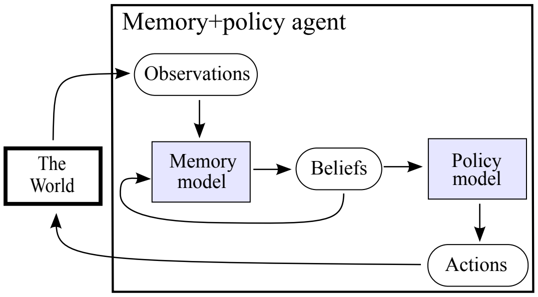

Fig. 5. A simple architecture that is sufficient for general cognition, but is not practical to train.

A diagram of a *memory+policy agent* is given in Figure 5. This architecture extends the policy agent with memory. It uses two models, one to update its internal beliefs from new observations, and one that maps from its current beliefs to actions. This architecture can be shown to be theoretically sufficient for general cognition. If its belief vector is sufficiently large, then it can represent any state that might occur in its world. (This may not always be practical, but we are currently evaluating only the theoretical sufficiency of this architecture.) If its memory model is implemented with an arbitrary function approximator, then it can theoretically update its beliefs as well as any other architecture could from the given observations. If its policy model is implemented with an arbitrary function approximator, then it can theoretically choose actions as well as any other architecture could from accurate beliefs. Therefore, we can say that the memory+policy architecture is theoretically sufficient for general cognition.

Since the memory+policy model is theoretically sufficient for general cognition, it would generally not be reasonable to resort to greater complexity for the purpose of trying to achieve new theoretical capabilities. However, the memory+policy architecture also has a significant practical limitation: it is very difficult to train. It requires a teacher that can unambiguously tell it which actions to choose in every region of its belief space. Even if a teacher is available to provide such thorough training, a memory+policy agent would never be able to exceed the capabilities of its teacher, which renders it of limited practical value.

The MANIC architecture adds some additional complexity so that it can separate its learning system from its decision-making system. This separation enables it to learn from unlabeled observations. That is, it can refine the learning system's ability to accurately anticipate the consequences of actions whether or not the teacher is available. With MANIC, supervision is only needed to guide its priorities, not its beliefs or choices. Yet, while being much more practical to train, MANIC is also provably sufficient for general cognition.

If we allow its belief vector, $\mathbf{v}_t$, to be arbitrarily large, then it it is sufficient to encode a correct representation of the agent's world, no matter how complex that world may be. If the transition function, f, is implemented with an arbitrary function approximator, then it is theoretically sufficient to correctly anticipate any state transitions. And, if the decoder, g, is also implemented with an arbitrary function approximator, then it will be able to accurately anticipate observations from correct beliefs. If we allow the genetic algorithm of the planning system to utilize an arbitrarily large population, then it approximates an exhaustive search through all possible plans, which will find the optimal plan. Since the action it chooses is always the first step in an optimal plan, its actions will be optimal for maximizing contentment. Finally, if the contentment model, h, is also implemented with a universal function approximator, then it is sufficient to approximate the ideal utility metric for arbitrary problems. Therefore, MANIC is sufficient for general cognition. In the next Section, we discuss additional details about why MANIC is also practical for training.

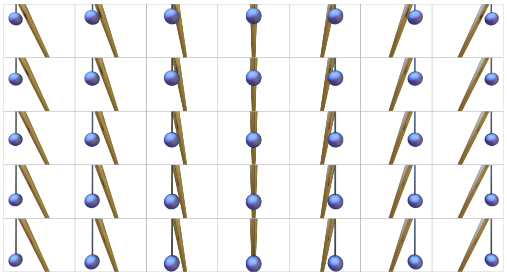

Fig. 6. A system consisting of a simulated crane viewed through a camera. Each observation consists of 9216 values, but the system itself only exhibits 2 degrees of freedom, so these images lie on a 2-dimensional non-linear manifold embedded in 9216-dimensional space. This figure depicts the entire manifold as represented by uniform sampling over its nonlinear surface.

IV. Training

Perhaps, the most significant insight for making MANIC practical for training comes from the high-level division between the learning system and the decision-making system. Because it divides where machine learning and artificial intelligence typically separate, well-established training methods become applicable, whereas such methods cannot be used with other cognitive architectures. Because the learning system does not choose candidate actions, but only anticipates their effect, it can learn from every observation that the agent makes, even if the actions are chosen randomly, and even when no teacher is available.

The learning system in MANIC is a recurrent architecture because the current beliefs, $\mathbf{v}_t$, are used by the transition model to anticpate subsequent beliefs, $\mathbf{v}_{t+1}$. Recurrent architectures are notoriously difficult to train, due to the chaotic effects of feedback that result from the recurrence [21], [22], [23], [24], [25]. This tends to create both large hills and valleys throughout the error surface, so local optimization methods must use very small learning rates. Further, local optimization methods are susceptible to the frequent local optima that occur throughout the error space of recurrent architectures, and global optimization methods tend to be extremely slow.

However, recent advances in nonlinear dimensionality reduction provide a solution for cases where observations lie on a high-dimensional manifold. Significantly, this situation occurs when observations consist of images that derive from continuous space. In other words, MANIC can handle the case of a robot equipped with digital cameras, which has obvious analogy with humans equipped with optical vision.

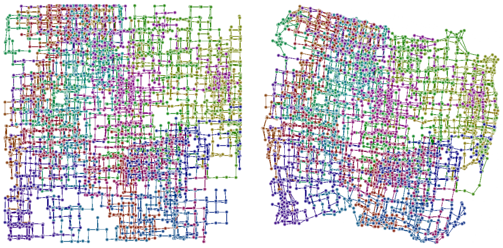

Fig. 7. **Left:** The hidden states visited in a random walk with a simulated crane system. Color is used to depict time, starting with red and ending with purple. The horizontal axis shows boom position, and the vertical axis shows cable length. The model was shown images of the crane, but was not allowed to view the hidden state. **Right:** Estimated states calculated by reducing the dimensionality of observed images. Although differing somewhat from the actual states, these estimates were close enough to bootstrap training of a recurrent model of system dynamics.

When observations are high-dimensional, a good initial estimate of state (or in this case, beliefs) can be obtained by reducing the dimensionality of those observations. Co-author Gashler demonstrated this approach in 2011 with a method that trained deep recurrent artificial neural networks to model dynamical systems [26]. For example, consider the crane system depicted in Figure 6. We used images containing 64×48 pixels in 3 color channels, for a total of 9216 values. We performed a random walk through the state space of the crane by moving the crane left, right, up, or down, at random to collect observations. (To demonstrate robustness, random noise was injected into state transitions as well as the observed images.) Using a nonlinear dimensionality reduction technique, we reduced the 9216-dimensional sequence of observed images down to just 2 dimensions (because the system has only 2 degrees of freedom) to obtain an estimate of the state represented in each high-dimensional image. (See Figure 7.) Significant similarity can be observed between the actual (left) and estimated (right) states. Consequently, this approach is ideal for bootstrapping the training of a recurrent model of system dynamics. When the beliefs are initialized to reasonable intial values, local optima is much less of a problem, so regular stochastic gradient descent can be used to refine the model from subsequent observations.

In the context of MANIC, this implies that nonlinear dimensionality reduction can be used to estimate each $\mathbf{v}_t$. Then, g can be trained to map from each $\mathbf{v}_t$ to $\mathbf{x}_t$, g^+ can be trained to map from each $\mathbf{x}_t$ to $\mathbf{v}_t$, and f can be trained to map from each $\mathbf{v}_t$ to $\mathbf{v}_{t+1}$. Note that each of these mappings depends on having a reasonable estimate of $\mathbf{v}_t$. These values are not typically known with recurrent architectures, but digital images provide sufficient information that unsupervised dimensionality reduction methods can estimate $\mathbf{v}_1, \mathbf{v}_2, \cdots$ from $\mathbf{x}_1, \mathbf{x}_2, \cdots$ very well. When an estimate for each $\mathbf{v}_t$ is known, training the various components of the learning system reduces to a simple supervised learning problem.

A similar three-step approach can be used to bootstrap the learning system of MANIC:

- Gather high-dimensional observations.
- Reduce observations to an initial estimate of beliefs.
- Use beliefs to train the transition model.

The last step, training the transition model, is difficult with recurrent models because gradient-based methods tend to get stuck in local optima. However, because dimensionality reduction can estimate beliefs prior to training, it reduces to a simple supervised training approach. It has been shown that bootstrapping the training of neural networks can effectively bypass the local optima that otherwise cause problems for refining with gradient-based approaches [27]. This effect is much more dramatic with recurrent neural networks, since they create local optima throughout their model space [21]. Thus, after initial training, the system can be maintained using regular backpropagation to refine the learning system from subsequent observations.

With the crane dynamical system, we were able to accurately anticipate dynamics several hundred time-steps into the future [26], even with injected noise. To validate the plausibility of our planning system, we also demonstrated this approach on another problem involving a robot that navigates within a warehouse. We first trained the learning system on a sequence of random observations. Then, using only observations predicted by the learning system, we were able to successfully plan an entire sequence of actions by using the learning system to anticipate the "beliefs" of MANIC. The path we planned is shown in Figure 8.B. We, then, executed this plan on the actual system. The actual states through which it passed are shown in Figure 8.C. Even though MANIC represented its internal belief states differently from the actual system, the anticipated observations were very close to the actual observations.

Many other recent advances in dimensionality reduction with deep artificial neural networks validate that this general approach is effective for producing internal intrinsic representations of external observations.[28], [29], [30], [31], [32].

The decision-making system contains one model, h, which needs to be trained to learn what constitutes homeostasis (or "contentment") for the system. This is done using a type of reinforcment learning. Because motivations are subjectively tied to human preferences, the motivations for an agent that humans would receive as intellgent necessarily depends on human teaching. Therefore, we assume that a human teacher is periodically available to direct the MANIC agent. In cases where no human teacher is available, the contentment model could also be trained using a competitive or evolutionary approach. This is done by instantiating multiple MANIC agents with variations in their contentment functions, and allowing the more fit instantiations to survive.

When MANIC makes plans, it utilizes its learning system to convert each plan from a sequence of actions to a corresponding video of anticipated observations. In many ways, these videos of anticipated observations may be analogous with the dreams or fantasies that humans produce internally as they sleep or anticipate future encounters. Although differences certainly exist, this similar side-effect may indicate that the architecture within the human brain has certain similarities with MANIC. Ironically, the imagination of the artificial system is more accessible than that of biological humans, enabling humans to examine the inner imaginings of the artificial system more intimately than they can with each other.

The videos of imagined observations are presented to the

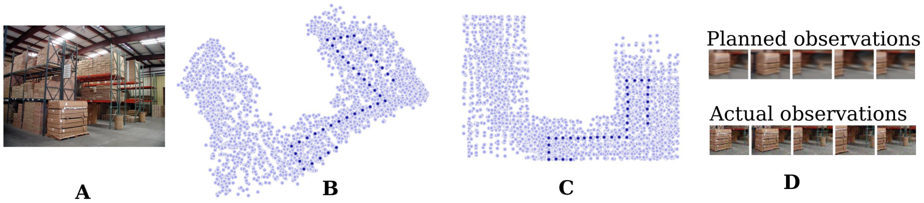

Fig. 8. **A:** A model was trained to capture the dynamics of a robot that uses a camera to navigate within a warehouse. **B:** A sequence of actions was planned entirely within the belief-space of the model. **C:** When executed, the plan resulted in actual behavior corresponding with the agent's anticipated beliefs. **D:** A comparison of the agent's anticipated observations with those actually obtained when the plan was executed.

human teacher (when he or she is available) for consideration. The human, then, ranks these videos according to the disirability of the anticipated outcome. Note that these plans need not actually be executed to generate the corresponding video. Rather, the teacher only ranks imagined scenes. The precision with which the imagination of the agent corresponds with reality when it actually executes a plan depends only on the learning system (which is continually refined), and does not depend on the teacher. Because the teacher is only needed to refine the contentment model, only a reasonable amount of human attention is ever needed.

MANIC encapsulates a diversity of learning paradigms. The observation model, g, is trained in an unsupervised manner from camera-based observations. The transition model, f, and the observation model, g and g^+, are trained by supervised learning. The contentment model, h, is trained by reinforcement from a human oracle. In 1999, Doya identified anatomical, physiological, and theoretical evidence to support the hypotheses that the cerebellum specializes in supervised learning, the cerebral cortex specializes in unsupervised learning, and the basal ganglia specialize in reinforcement learning [33]. This suggests a possible correlation between the components of MANIC and those in the brain. Finally, the planning module in MANIC ties all of its pieces together to make intelligent plans by means of a genetic algorithm. We consider it to be a positive property that the MANIC architecture unifies such a wide diversity of learning techniques in such a natural manner. As each of these methods exists for the purpose of addressing a particular aspect of cognition, we claim that a general cognitive architecture must necessarily give place to each of the major paradigms that have found utility in artificial intelligence and machine learning.

V. Sentience through self perception

Sentience is a highly subjective and ill-defined concept, but is a critical aspect of the human experience. Consequently, it has been the subject of much focus in cognitive architectures. We propose a plausible theory that might explain sentience, and show that MANIC can achieve functional equivalence.

It is clear that MANIC *perceives* its environment, because it responds intelligently to the observations it makes. Its perception is implemented by its beliefs, which describe its understanding of the observations it makes, as well as its observation model, which connects its beliefs to the world. The term "awareness" is sometimes used with humans to imply a higher level of perception. It is not clear whether MANIC achieves "awareness", because the difference between "awareness" and "perception" is not yet well-defined. However, because we have proven that MANIC is sufficient for general cognition, we can say for sure that MANIC achieves something functionally equivalenct with awareness. That is, we know it can behave as if it is aware, but we do not know whether human awareness requires certain immeasurable properties that MANIC lacks.

Similarly, the term "sentience" contains aspects that are not yet well-defined, but other aspects of sentience are well-established. Specifically, sentience requires that an agent possess feelings that summarize its overall well-being, as well as an awareness of those feelings, and a desire to act on them. An agent that implements the well-defined aspects of sentience can be said to implement something functionally equivalent.

Our definition of sentience is best expressed as an analogy with perception: Perception requires the ability to observe the environment, beliefs that summarize or represent those observations, and a model to give the beliefs context. Similarly, we propose that sentience requires the ability to make introspective observations, "feelings" that summarize or represent them, and a model to give the feelings context. In other words, if sentience arises from self-awareness, then MANIC can achieve something functionally equivalent through self perception. In addition to a camera that observes the environment, MANIC can be equipped with the ability to observe its own internal state. (For example, it might be enabled to additionally observe the weights of its three models, f, g, and h, and its belief vector, $\mathbf{v}$.) Since MANIC is already designed for operating with high-dimensional observations, these introspective observations could simply be concatenated with the external observations it already makes. This would cause MANIC to utilize a portion of $\mathbf{v}$ to summarize its introspective observations.

Thus, $\mathbf{v}$ would represent both the "beliefs" and "feelings" of a MANIC agent. Its planning system would then implicilty make plans to maintaining homeostasis in both its beliefs and feelings. And its observation model would give context to its feelings by mapping between feelings and introspective observations, just as it does between beliefs and external observations. This theory of sentience is plausible with humans, because humans plan with regard to both their feelings and beliefs as a unified concept, maintaining both their external objectives and internal well-being, and because a well-connected brain, which humans have, would be sufficient to provide the "introspective observations" necessary to facilitate it.

Since MANIC learns to model its priorities from a human teacher in its contentment function, h, it will learn to give appropriate regard to its own feelings when they are relevant for its purpose. Presumably, this will occur when the human teacher directs it to maintain itself. Using the same observation model with both introspective and external observations, and using the same vector to model both feelings and beliefs, are both plausible because these design choices will enable MANIC to entangle its feelings with its environment, behavior that humans are known to exhibit.

VI. Conclusion

We presented a cognitive architecture called MANIC. This architecture unifies a diversity of techniques in the subdisciplines of machine learning and artificial intelligence without introducing much novelty. Yet, while relying on existing methods, and with minimal complexity, MANIC is a powerful cognitive architecture. We showed that it is sufficiently general to accomplish arbitrary cognitive tasks, and that it can be practically trained using recent methods.

We supported MANIC's design by referring to existing works that validate its individual components, and we made theoretical arguments about the capabilities that should emerge from combining them in the manner outlined by MANIC. The primary contribution of this paper is to show that these existing methods can already accomplish more of cognitive intelligence than is generally recognized. Our ultimate intent is to argue that if general intelligence is one of the ultimate objectives of the fields of machine learning and artificial intelligence, then they are very much on the right track, and it is not clear that any critical piece of understanding necessary for implementing a rudimentary consciousness is definitely missing.

References

[1] J. McCarthy, M. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955," *AI Magazine*, vol. 27, no. 4, pp. 12–14, 2006.

[2] H. A. Simon, *The Shape of Automation for Men and Management*. New York: Harper and Row, 1967.

[3] D. Cervier, *AI: The Tumultuous Search for Artificial Intelligence*. New York: Basic Books, 1993.

[4] P. McCorduck, *Machines Who Think*, 2nd ed. Natick, MA: A. K. Peters, Ltd., 2004.

[5] X. Gao, X. Gao, and S. Ovaska, "A modified elman neural network model with application to dynamical systems identification," in *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 1376–1381.

[6] S. Chen and S. Billings, "Representations of non-linear systems: the narmax model," *International Journal of Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[7] S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Kalman filtering and neural networks*. Wiley Online Library, 2001.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard?" *PLoS computational biology*, vol. 4, no. 1, p. e27, 2008.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] K.-S. Oh and K. Jung, "Gpu implementation of neural networks," *Pattern Recognition*, vol. 37, no. 6, pp. 1311–1314, 2004.

[12] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[14] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *Neural Networks, IEEE Transactions on*, vol. 8, no. 1, pp. 98–113, 1997.

[15] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE, 2010, pp. 253–256.

[16] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2011, pp. 1918–1921.

[17] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ, USA: Princeton University Press, 1961.

[18] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, vol. 25, 1995.

[19] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[20] G. Cybenko", "Continuous valued neural networks with two hidden layers are sufficient," Tufts University, Medford, MA, Tech. Rep., 1998.

[21] M. P. Cuéllar, M. Delgado, and M. C. Pegalajar, "An application of non-linear programming to train recurrent neural networks in time series prediction problems," *Enterprise Information Systems VII*, pp. 95–102, 2006.

[22] E. Sontag, "Neural networks for control," *Essays on Control: Perspectives in the Theory and its Applications*, vol. 14, pp. 339–380, 1993.

[23] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, pp. 1691–1724, 1995.

[24] D. Floreano and F. Mondada, "Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot," *From animals to animats*, vol. 3, pp. 421–430, 1994.

[25] A. Blanco, M. Delgado, and M. C. Pegalajar, "A real-coded genetic algorithm for training recurrent neural networks," *Neural Networks*, vol. 14, no. 1, pp. 93–105, 2001.

[26] M. S. Gashler and T. R. Martinez, "Temporal nonlinear dimensionality reduction," in *Proceedings of the International Joint Conference on Neural Networks*. IEEE Press, 2011, pp. 1959–1966.

[27] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[28] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[29] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[30] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 833–840.

[31] Y. Bengio, A. C. Courville, and J. S. Bergstra, "Unsupervised models of images by spike-and-slab rbms," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1145–1152.

[32] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[33] K. Doya, "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?" *Neural networks*, vol. 12, no. 7, pp. 961–974, 1999.