Review Paper

A thousand brains: toward biologically constrained AI

Kjell Jørgen Hole[1] · Subutai Ahmad[2]

Received: 3 February 2021 / Accepted: 25 June 2021
Published online: 20 July 2021

Abstract

This paper reviews the state of artificial intelligence (AI) and the quest to create general AI with human-like cognitive capabilities. Although existing AI methods have produced powerful applications that outperform humans in specific bounded domains, these techniques have fundamental limitations that hinder the creation of general intelligent systems. In parallel, over the last few decades, an explosion of experimental techniques in neuroscience has significantly increased our understanding of the human brain. This review argues that improvements in current AI using mathematical or logical techniques are unlikely to lead to general AI. Instead, the AI community should incorporate neuroscience discoveries about the neocortex, the human brain's center of intelligence. The article explains the limitations of current AI techniques. It then focuses on the biologically constrained *Thousand Brains Theory* describing the neocortex's computational principles. Future AI systems can incorporate these principles to overcome the stated limitations of current systems. Finally, the article concludes that AI researchers and neuroscientists should work together on specified topics to achieve biologically constrained AI with human-like capabilities.

Keywords Limitations of narrow AI · General AI · Biologically constrained general AI · Common cortical algorithm · Neuroscience · Neocortex

1 Introduction

Artificial intelligence (AI) is the study of techniques that allow computers to learn, reason, and act to achieve goals [1]. Recent AI research has focused on creating *narrow* AI systems that perform one well-defined task in a single domain, such as facial recognition, Internet search, driving a car, or playing a computer game. AI research's long-term goal is to create *general* AI with human-like cognitive capabilities. Whereas narrow AI performs a single cognitive task, general AI would perform well on a broad range of cognitive challenges.

Machine learning is an essential type of narrow AI. It permits machines to learn from big data sets without being explicitly programmed. At the time of this writing (summer 2021), the international AI community focuses on a type of machine learning called deep learning. It is a family of statistical techniques for classifying patterns using artificial neural networks with many layers. Deep learning has produced breakthroughs in image and speech recognition, language translation, navigation, and game playing [2, Ch. 1].

This tutorial-style review assesses the state of AI research and its applications. It first compares human intelligence and AI to understand what narrow AI systems, especially deep learning systems, can and cannot do. The paper argues that today's mathematical and logical approaches to AI have fundamental limitations preventing them from attaining general intelligence even remotely close to human beings. In other words, the non-biological

✉ Kjell Jørgen Hole, hole@simula.no; Subutai Ahmad, sahmad@numenta.com | [1]Director, Simula UiB, Thormøhlens gate 53D, 5006 Bergen, Norway. [2]VP of Research and Engineering, Numenta, 791 Middlefield Road, Redwood City, CA 94063, USA.

path of narrow AI does not lead to intelligent machines that understand and act similarly to humans.

Next, the article concentrates on the biological path to general AI offered by the field of neuroscience. The review focuses on the *Thousand Brains Theory* by Jeff Hawkins and his research team (numenta.com). The theory uncovers the computational principles of the neocortex, the human brain's center of intelligence. The review outlines a computational model of intelligence based on these neocortical principles. The model shows how future AI systems can overcome the stated limitations of current deep learning systems. Finally, the review summarizes essential insights from the neocortex and suggests cooperation between neuroscientists and AI researchers to create general AI.

The review article caters to a broad readership, including students and practitioners, with expertise in AI or neuroscience, but not both areas. The main goal is to introduce AI experts to neuroscience (Sects. 4–6). However, we also want to introduce neuroscientists to AI (Sects. 2 and 3) and create a shared understanding, facilitating cooperation between the areas (Sect. 7). Since the relevant literature is vast, it is impossible to reference all articles, books, and online resources. Therefore, rather than adding numerous citations to catalog the many AI and neuroscience researchers' contributions, the authors have prioritized well-written books, review articles, and tutorials. Together, these sources and their references constitute a starting point for readers to learn more about the intersection between AI and neuroscience and the reverse engineering of the human neocortex.

2 Types of intelligence

This section introduces and contrasts human intelligence with narrow and general AI.

2.1 Human intelligence

Human intelligence is the brain's ability to learn a model of the world and use it to understand new situations, handle abstract concepts, and create novel behaviors, including manipulating the environment [3, 4]. The brain and the body are massively and reciprocally connected. The communication is parallel, and information flows both ways. The philosophers George Lakoff and Mark Johnson [5, 6] have emphasized the importance of the brain's *sensorimotor integration*, i.e., the integration of sensory processing and generation of motor commands.

It is the sensorimotor mechanisms in the brain that allow people to perceive and move around. Sensorimotor integration is also the basis for abstract reasoning. Humans have so-called *embodied reasoning* because the sensorimotor mechanisms shape their abstract reasoning abilities. The structure of the brain and body, and how it functions in the physical world, constrains and informs the brain's capabilities [5, Ch. 3].

2.2 AI for computers

AI researchers have traditionally favored mathematical and logical rather than biologically constrained approaches to creating intelligence [7, 8]. In the past, classical or symbolic AI applications, like expert systems [9] and game playing programs, deployed explicit rules to process high-level (human-readable) input symbols. Today, AI applications use artificial neural networks to process vectors of numerical input symbols. In both cases, an AI program running on a computer processes input symbols and produces output symbols. However, unlike the brain's sensorimotor integration and embodied reasoning, current AI is almost independent of the environment. The AI programs run internally on the computer without much interaction with the world through sensors.

2.3 Learning algorithms

A typical AI system (with a neural network) analyzes an existing set of data called the *training set*. The system uses a *learning algorithm* to identify patterns and probabilities in the training set and organizes them in a model that generates outputs in the form of classifications or predictions. The system can then run new data through the model to obtain answers to questions such as "what should be the next move in a digital game?" and "should a bank customer receive a loan?" There are three broad categories of learning algorithms in AI, where the first two use static training sets and the third uses a fixed environment.

Supervised learning The training set contains examples of input data and corresponding desired output data. We refer to the training set as *labeled* since it connects inputs to desired outputs. The learning algorithm's goal is to develop a mapping from the inputs to the outputs.

Unsupervised learning The training set contains only input data. The learning algorithm must find patterns and features in the data itself. The aim is to uncover hidden structures in the data without explicit labels. Unsupervised learning is more complicated and less mature than supervised learning.

Reinforcement learning The training regime consists of an agent taking actions in a fixed artificial environment and receiving occasional rewards. The goal of the learning algorithm is to make optimal actions based on these rewards. One of the first successful examples of reinforcement learning was the TD-Gammon program

[10], which learned to play expert-level backgammon. TD-gammon played hundreds of thousands of games and received one reward (win/lose) for each game.

The differences between the three learning types are not always obvious, especially when multiple methods are combined to train a single system. An important distinction is whether a learning algorithm continues to learn after the initial training phase. Typical AI systems do not learn after training, no matter the type of learning algorithm they use. Once training is complete, the systems are frozen and rigid. Any changes require retraining the entire system from scratch. The ability to keep learning after training is often called continuous learning.

2.4 Narrow and general AI

We discriminate between narrow and general AI. Narrow AI is a set of mathematical techniques typically using fixed training sets to generate classifications or predictions. Each narrow AI system performs one well-defined task in a single domain. The best narrow or single-task AI systems outperform humans. However, most narrow AI systems must retrain with new training sets to learn other tasks. The ultimate goal of general AI, also called artificial general intelligence, is to achieve or surpass human-level performance on a broad range of tasks and learn new ones on the fly. General AI systems have common sense knowledge, adapt quickly to new situations, understand abstract concepts, and flexibly use their knowledge to plan and manipulate the environment to achieve goals. This paper does not assume that general AI requires human-like subjective experiences, such as pain and happiness, but see [11] for a different view.

3 Status of AI

The human brain's limits are due to the biological circuits' slow speed, the limited energy provided by the body, and the human skull's small volume. Artificial systems have access to faster circuits, more energy, and nearly limitless short- and long-term memory with perfect recall [1]. Whereas the embodied brain can only learn from data received through its biological senses (such as sight and hearing), there is nearly no limit to the sensors an AI system can utilize. A distributed AI system can simultaneously be in multiple places and learn from data not available to the human brain. Although artificial systems are not limited to brain-like processing of sensory data from natural environments, they have not achieved general intelligence. This section discusses the strengths and limitations of AI systems as compared to human intelligence.

3.1 Strengths of narrow AI

We may associate both a natural and a human-made system with a big set of discrete states. A narrow AI system running on a fast computer can explore more of this state space than the human brain to determine desirable states. Furthermore, machines have the processing power to be more precise and the memory to store everything. These are the reasons why machine learning can outperform humans in bounded, abstract domains [2, Ch. 1]. Whereas most such AI programs require vast labeled training sets generated by humans, one fascinating program, AlphaZero [12], achieved superhuman performance using reinforcement learning and Monte Carlo tree search. AlphaZero trained to play the board games Go, chess, and shogi without any human input (apart from the basic rules of chess) by playing many games against itself.

Other examples of AI-based applications that outperform humans are warehouse robots using vision to sort merchandise and security programs detecting and mitigating a large number of attacks [1]. Whereas a human learns mostly from experience, which is a relatively slow process, a narrow AI system can make many copies of itself to simultaneously explore a broad set of possibilities. Often, these systems can discard adverse outcomes with little cost.

AI and human cooperation in human-in-the-loop systems have enormous possibilities because narrow AI augments people's ability to organize data, find hidden patterns, and point out anomalies [13]. For example, machine translation can produce technically accurate texts, but human translators must translate idioms or slang correctly. Likewise, narrow AI helps financial investors determine what and when to trade, improves doctors' diagnoses of patients, and assists with predictive maintenance in many areas by detecting anomalies.

AI applications could learn together with people and provide guidance. Novices entering the workforce might get AI assistants that offer support on the job. Although massive open online courses (MOOCs) need human teachers to encourage and inspire undergraduates, the MOOCs might use AI to personalize and enhance student feedback. AI could determine what topics the students find hard to learn and suggest how to improve the teaching. In short, humans working with narrow AI programs may be critical to solving challenging problems in future.

3.2 Limitations of narrow AI

AlphaZero and all other narrow AI programs do not know what they do. They cannot transfer their performance to other domains and could not even play tic-tac-toe without a redesign and extensive practice. Furthermore, creating a narrow AI solution with better-than-human performance requires much engineering work, including selecting the best combination of training algorithms, adjusting the training parameters, and testing the trained system [14]. Even with reinforcement learning, narrow AI systems are proxies for the people who made them and the specific training environments.

Most learning algorithms, particularly deep learning algorithms, are *greedy*, *brittle*, *rigid*, and *opaque* [15, 16]. The algorithms are greedy because they demand big training sets; brittle because they frequently fail when confronted with a mildly different scenario than in the training set; rigid because they cannot keep adapting after initial training; and opaque since the internal representations make it challenging to understand their decisions. In practice, deep learning systems are black boxes to users. These shortcomings are all serious, but the core problem is that all narrow AI systems are *shallow* because they lack abstract reasoning abilities and possess no common sense about the world.

It can be downright dangerous to allow narrow AI solutions to operate without people in the loop. Narrow AI systems can make serious mistakes no sane human would make. For example, it is possible to make subtle changes to images and objects that fool machine learning systems into misclassifying objects. Scientists have attached stickers to traffic signs [17], including stop signs, to fool machine learning systems into misclassifying them. MIT students have tricked an AI-based vision system into wrongly classifying a 3D-printed turtle as a rifle [18]. The susceptibility to manipulation is a big security issue for products that depend on vision, especially self-driving cars. (See Meredith Broussard's overview [19, Ch. 8] of the obstacles to creating self-driving cars using narrow AI.) While AI solutions do not make significant decisions from single data points, more work is needed to make AI systems robust to deliberate attacks.

3.3 There is no general AI (Yet)

A general AI system needs to be a wide-ranging problem solver, robust to obstacles and unwelcome surprises. It must learn from setbacks and failures to come up with better strategies to solve problems. Humans integrate learning, reasoning, planning, and communication skills to solve various challenging problems and reach common goals. People learn continuously and master new functions without forgetting how to perform earlier mastered tasks. Humans can reason and make judgments utilizing contextual information way beyond any AI-enhanced device. People are good at idea creation and innovative problem solutions—especially solutions requiring much sensorimotor work or complex communication. No artificial entity has achieved this general intelligence. In other words, general AI does not yet exist.

4 The biological path to general AI

The international AI community struggles with the same challenging problems in common sense and abstract reasoning as they did six decades ago [15, 20]. However, lately, some AI researchers have realized that neuroscience could be essential to create intelligent machines [21]. This section introduces the field of computational neuroscience and the biological path to general AI.

4.1 Computational neuroscience

Neuroscience has long focused on experimental techniques to understand the fundamental properties of brain cells, or neurons, and neural circuits. Neuroscientists have conducted many experiments, but less work successfully distills principles from this vast collection of results. A theoretical framework is needed to understand what the results tell us about the brain's large-scale computation. Such a framework must allow scientists to create and test hypotheses that make sense in the context of earlier findings.

The book by Peter Sterling and Simon Laughlin [3] goes a long way toward making sense of many neuroscience findings. Their theoretical framework outlines organizing principles to explain why the brain computes much more efficiently than computers. Although the framework helps to understand the brain's low-level design principles, it provides little insight into the algorithmic or high-level computational aspects of human intelligence. We, therefore, need an additional framework to determine the brain's computational principles.

Computational neuroscience is a branch of neuroscience that employs experimental data to build mathematical models and carry out theoretical analyses to understand the principles that govern the brain's cognitive abilities. Since the field focuses on biologically plausible models of neurons and neural systems, it is not concerned with biologically unrealistic disciplines such as machine learning and artificial neural networks. Patricia S. Churchland and Terrence J. Sejnowski addressed computational neuroscience's foundational ideas in a classic book from 1992 [22].

Sejnowski published another book [2] in 2018, documenting that the deep learning networks are biologically inspired, but not biologically constrained. These narrow AI systems need many more examples to recognize new objects than humans, suggesting that reverse engineering of the brain could lead to more efficient biologically constrained learning algorithms. To carry out this reverse

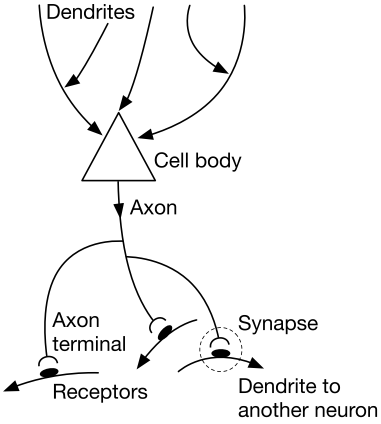

Fig. 1 Neuron with multiple dendrites and one axon. A synapse consists of an axon terminal, a cleft, and receptors on a dendrite of another neuron

engineering, we need to view the brain as a hierarchical computational system consisting of multiple levels, one on top of the other. A level creates new functionality by combining and extending the functionality of lower levels [4].

We focus on a specific area of the brain known as the neocortex, the primary brain area associated with intelligence. The neocortex is an intensely folded sheet with a thickness of about 2.5 mm. When laid out flat, it has the size of a formal dinner napkin. The neocortex constitutes roughly 70 percent of the brain's volume and contains more than 10 billion neurons (brain cells). A typical neuron (Fig. 1) has one tail-like axon and several treelike extensions called dendrites. When a cell fires, an electrochemical pulse or spike travels down the axon to its terminals.

A signal jumps from an axon terminal to the receptors on a dendrite of another neuron. The axon terminal, the receptors, and the cleft between them constitute a *synapse*. The axon terminal releases neurotransmitters into the synaptic cleft to signal the dendrite. Thus, the neuron is a signaling system where the axon is the transmitter, the dendrites are the receivers, and the synapses are the connectors between the axons and dendrites. Neurons in the neocortex typically have between 1,000 and 20,000 synapses.

The neocortex consists of regions engaged in cognitive functions. As we shall see, the neocortical regions realizing vision, language, and touch operate according to the same principles. The sensory input determines a region's purpose. The neocortex generates body movements, including eye motions, to change the sensory inputs and learn quickly about the world.

At the time of this writing, there exist many efforts in computational neuroscience to reverse engineer mammalian brains, especially to understand the computational principles of the neocortex [23]. Considerable reverse engineering efforts include the US government program *Machine Intelligence from Cortical Networks*, the Swiss *Blue Brain Project*, and the *Human Brain Project* funded mainly by the EU. However, so far, these projects have not assimilated their research results into frameworks.

The research laboratory Numenta has shared a framework, the *Thousand Brains Theory* [24], based on known computational principles of the neocortex. This theory describes the brain's biology in a way that experts can apply to create general AI. We later focus on a particular realization of the *Thousand Brains Theory* to illuminate the path to biologically constrained general AI. Jeff Hawkins' first book [25] describes an early version of the theory to non-experts. His second book [26] contains an updated theory description for the non-specialist.

4.2 Biological constraints

To understand why we consider the biological path toward general AI, we observe that the limited progress toward intelligent systems after six decades of research strongly indicates that few investigative paths lead to general AI. Many paths seem to lead nowhere. Since guidance is needed to discover the right approach in a vast space of algorithms containing few solutions, the AI community should focus on the only example we have of intelligence, namely the brain and especially the neocortex. Although it is possible to add mathematical and logical methods to biologically plausible algorithms based on the neocortex, the danger is that non-biological methods cause scientists to follow paths that do not lead to general AI.

This article's premise is that, until the AI community deeply understands the nature of intelligence, we treat biological plausibility and neuroscience constraints as strict requirements. Using neuroscience findings, Hawkins argues [25–27] that continued work on today's narrow AI techniques cannot lead to general AI because the techniques are missing necessary biological properties. We consider six properties describing the "data structures" and "architecture" of the neocortex, starting with three data structure properties:

Sparse data representations A computer uses dense binary vectors of 1s and 0s to represent types of data (ASCII being an example). A deep learning system uses dense vectors of real numbers where a large fraction of the elements are nonzero. These representations are in stark contrast to the highly sparse representations in the neocortex, where at a point in time, only a small percentage of the values are nonzero. Sparse representations allow general AI systems to efficiently represent and process data in a brain-like manner robust to changes caused by internal errors and noisy data from the environment.

Realistic neuron model Nearly all artificial neural networks use very simple artificial neurons. The neocortex contains neurons with dendrites, axons, synapses, and dendritic processing. General AI systems based on the neocortex need a more realistic neuron model (Fig. 1) with brain-like connections to other neurons.

Reference frames General AI systems must be able to make predictions in a dynamic world with constantly changing sensory input. This ability requires a data structure that is invariant to both internally generated movements and external events. The neocortex uses reference frames to store all knowledge and has mechanisms that map movements into locations in these frames. General AI systems must incorporate models and computation based on movement through reference frames.

Next, we focus on three fundamental architectural properties of the neocortex:

Continuous online learning Whereas most narrow AI systems use offline, batch-oriented, supervised learning with labeled training data, the learning in the neocortex is unsupervised and occurs continuously in real time using data streaming from the senses. The ability to dynamically change and "rewire" the connections between brain cells is vital to realize the neocortex's ability to learn continuously [28].

Sensorimotor integration Body movements allow the neocortex to actively change its sensory inputs, quickly build models to make predictions, and detect anomalies, and, thus, perceive the physical and cultural environments. Similarly, general AI systems based on the neocortex must be embodied in the environment and actively move sensors to build predictive models of the world.

Single general-purpose algorithm The neocortex learns a detailed model of the world across multiple sensory modalities and at multiple levels of abstraction. As first outlined by Vernon Mountcastle [29], all neocortex regions are fundamentally the same and contain a repeating biological circuitry that forms the *common cortical algorithm*. (Note that there are variations in cell types, the ratio of cells, and the number of cell layers between the neocortical regions.) Understanding and implementing such a common cortical algorithm may be the only path to scalable general-purpose AI systems.

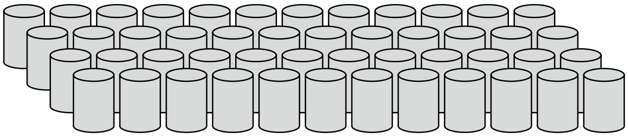

Fig. 2 Cortical columns in a region of the neocortex

5 Overview of the HTM model

The term *hierarchical temporal memory* (HTM) describes a specific realization of the *Thousand Brains Theory*. HTM builds models of physical objects and conceptual ideas to make predictions, and it generates motor commands to interact with the surroundings and test the predictions. The continuous testing allows HTM to update the predictive models and, thus, its knowledge, leading to intelligent behavior in an ever-changing world [25–27, 30].

This section first explains how HTM's general structure models the neocortex. It then discusses each HTM part in more detail. To provide a compact, understandable description of HTM, we simplify the neuroscience, keeping only essential information.

5.1 General structure

We first consider the building blocks of the neocortex, the neurons. Many neurons in the neocortex are excitatory, while others are inhibitory. When an excitatory neuron fires, it causes other neurons to fire. If an inhibitory neuron fires, it prevents other neurons from firing. The HTM model includes a mechanism to inhibit neurons, but it focuses on the excitatory neurons' functionality since about 80% of the neurons in the neocortex are excitatory [23, Ch. 3]. Because the pyramidal neurons [31] constitute the majority of the excitatory neurons, HTM contains abstract pyramidal neurons called *HTM neurons*.

The HTM model consists of regions of HTM neurons. The regions are divided into vertical *cortical columns* [29], as shown in Fig. 2. All cortical columns have the same laminar structure with six horizontal layers on top of each other (Fig. 3). Five of the layers contain *mini-columns* [32] of HTM neurons. A neuron in a mini-column connects to many other neurons in complicated ways (not shown in Fig. 3). A mini-column in the neocortex can span multiple layers. All cortical columns run essentially the same learning algorithm, the previously mentioned common cortical algorithm, based on their common circuitry.

The HTM regions connect in approximate hierarchies. Figure 4 illustrates two imperfect hierarchies of vertically connected regions with horizontal connections between the hierarchies. All regions in a hierarchy integrate sensory

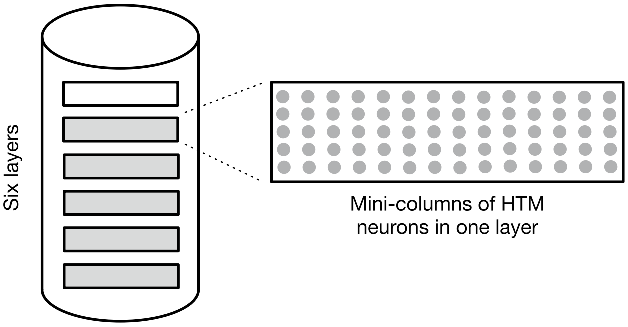

Fig. 3 Single cortical column in HTM with six layers, where five layers contain mini-columns of HTM neurons

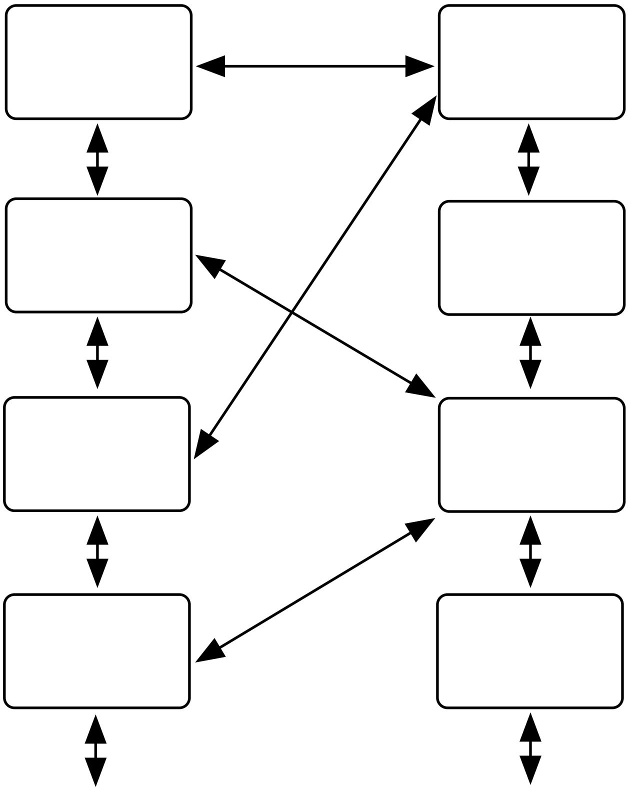

Fig. 4 Two approximate hierarchies of vertically connected regions with horizontal connections between the hierarchies

and motor information. Information flows up and down in a hierarchy and both ways between hierarchies.

Building on the general structure of HTM, the rest of the section explains how the HTM parts fulfill the previously listed data structure properties (sparse data representations, realistic neuron model, and reference frames) and the architectural properties (continuous online learning, sensorimotor integration, and single general-purpose algorithm). It also describes how the HTM parts depend on each other.

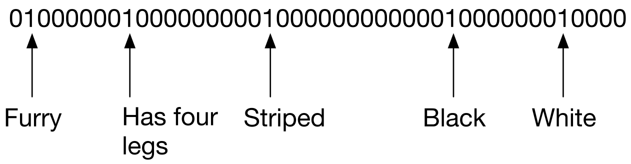

Fig. 5 Simplified SDR for a zebra. Adapted from [23]

5.2 Sparse distributed representations (SDRs)

Empirical evidence shows that every region of the neocortex represents information using sparse activity patterns made up of a small percentage of active neurons, with the remaining neurons being inactive. An SDR is a set of binary vectors where a small percentage of 1s represent active neurons, and the 0s represent inactive neurons. The small percentage of 1s, denoted the *sparsity*, varies from less than one percent to several percent. SDRs are the primary data structure used in the neocortex and used everywhere in HTM systems. There is not a single type of SDRs in HTM but distinct types for various purposes.

While a bit position in a dense representation like ASCII has no semantic meaning, the bit positions in an SDR represent a particular property. The semantic meaning depends on what the input data represents. Some bits may represent edges or big patches of color; others might correspond to different musical notes. Figure 5 shows a somewhat contrived but illustrative example of an SDR representing parts of a zebra. If we flip a single bit in a vector from a dense representation, the vector may take an entirely different value. In an SDR, nearby bit positions represent similar properties. If we invert a bit, then the description changes but not radically.

The mathematical foundation for SDRs and their relationship to the HTM model is described in [33–35]. SDRs are crucial to HTM. Unlike dense representations, SDRs are robust to large amounts of noise. SDRs allow HTM neurons to store and recognize a dynamic set of patterns from noisy data. Taking unions of sparse vectors in SDRs make it possible to perform multiple simultaneous predictions reliably. The properties described in [33, 34] also determine what parameter values to use in HTM software. Under the right set of parameters, SDRs enable a massive capacity to learn temporal sequences and form highly robust classification systems.

Every HTM system needs the equivalent of human sensory organs. We call them "encoders." A set of encoders allows implementations to encode data types such as dates, times, and numbers, including coordinates, into SDRs [36]. These encoders enable HTM-based systems to operate on other data than humans receive through their senses, opening up the possibility of intelligence in

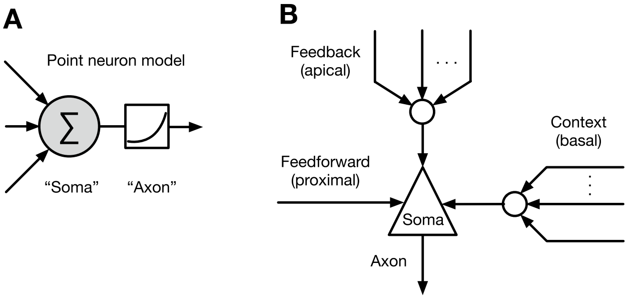

Fig. 6 **a** The point neuron is a simple model neuron that calculates a weighted sum of its inputs and passes the result through a nonlinearity. **b** Sketch of the HTM neuron adapted from [37]. The feedforward input determines whether the soma moves into the active state and fires a signal on the axon. The sets of feedback and context dendrites determine whether the soma moves into the predictive state

areas not covered by humans. An example is small intelligent machines operating at the molecular level; another is intelligent machines operating in toxic environments to humans.

5.3 HTM neurons

Biological neurons are pattern recognition systems that receive a constant stream of sparse inputs and send outputs to other neurons represented by electrical spikes known as action potentials [37]. Pyramidal neurons, the most common neurons in the neocortex, are pretty different from the typical neurons modeled in deep learning systems. Deep learning uses so-called *point neurons*, which compute a weighted sum of scalar inputs and send out scalar outputs, as shown in Fig. 6a. Pyramidal neurons are significantly more complicated. They contain separate and independent zones that receive diverse information and have various spatiotemporal properties [38].

Figure 6b illustrates how the HTM neuron models the structure of pyramidal neurons (the synapses and details about the signal processing are left out). The HTM neuron receives sparse input from artificial dendrites, segregated into areas called the *apical* (feedback signals from upper layers), *basal* (signals within the layer), and *proximal* (feedforward input) integration zones. Each dendrite in the apical and basal integration zones is an independent processing entity capable of recognizing different patterns [37].

In pyramidal neurons, when one or more dendrites in the apical or basal zone detect a pattern, they generate a voltage spike that travels to the pyramidal cell body or soma. These dendritic spikes do not directly create an action potential but instead cause a temporary increase in the cell body's voltage, making it primed to respond quickly to subsequent feedforward input. The HTM neuron models these changes using one of three states: active, predictive, or inactive. In the active state, the neuron outputs a 1 on the artificial axon analogous to an action potential; in the other states, it outputs a 0. Patterns detected on the proximal dendrite drive the HTM neuron into the active state, representing a natural pyramidal neuron firing. Pattern matching on the basal or apical dendrites moves the neuron into the predictive state, representing a primed cell body that is not yet firing.

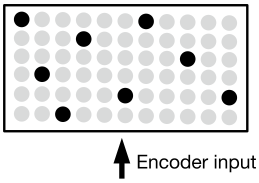

Fig. 7 A layer of HTM neurons organized into mini-columns. The black HTM neurons represent 1s and the rest represent 0s

5.4 Continuous online learning

To understand how HTM learns time-based sequences or patterns, we consider how the network operates before describing the connection-based learning. Consider a layer in a cortical column. Figure 7 depicts a layer of mini-columns with interconnected HTM neurons (connections not shown). This network receives a part of a noisy sequence at each time instance, given by a sparse vector from an SDR encoder. All HTM neurons in a mini-column receive the same subset of bits on their proximal dendrites, but different mini-columns receive different subsets of the vector. The mini-columns activate neurons, generating 1s, to represent the sequence part. (Here, we assume that the network has learned a consistent representation that removes noise and other ambiguities.)

Each HTM neuron predicts its activation, i.e., moves into the predictive state in various contexts by matching different patterns on its basal or apical dendrites. Figure 8 depicts dark gray neurons in the predictive state. According to this prediction, input on the feedforward dendrites will move the neuron into the active state in the following time instance. To illustrate how predictions occur, let a network's context be the states of its neurons at time $t-1$. Some neurons move to the predictive state at time t based on the previous context (Fig. 8a). If the context contains feedback projections from higher levels, the network forms predictions based on high-level expectations (Fig. 8b). In both cases, the network makes temporal predictions.

Learning occurs by reinforcing those connections that are consistent with the predictions and penalizing connections that are inconsistent. HTM creates new connections when predictions are mistaken [37]. When HTM first creates a network of mini-columns, it randomly generates

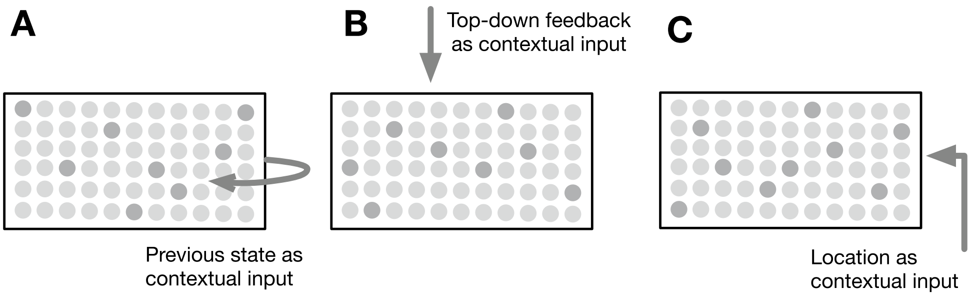
A
B
Top-down feedback as contextual input
C
Previous state as contextual input
Location as contextual input

Fig. 8 A layer of HTM neurons organized into mini-columns makes predictions represented by dark gray neurons. Predictions are based on context. **a** If the context represents the previous state of the layer, then the layer acts as a sequence memory. **b** If the context consists of top-down feedback, the layer forms predictions based on high-level expectations. **c** If the context consists of an allocentric (object-centrix) location signal, the layer makes sensory predictions based on location within an external reference frame

potential connections between the neurons. HTM assigns a scalar value called "permanence" to each connection. The permanence takes on values from zero to one. The value represents the longevity of a connection as illustrated in Fig. 9.

If the permanence value is close to zero, there is a potential for a connection, but it is not operative. If the permanence value exceeds a threshold, such as 0.3, then the connection becomes operative, but it could quickly disappear. A value close to one represents an operative connection that will last for a while. A (Hebbian-like) rule, using only information local to a neuron, increases and decreases the permanence value. Note that while neural networks with point neurons have fixed connections with real weights, the operative connections in HTM networks have weight one, and the inoperative connections have weight zero.

The whole HTM network operates and learns as follows. It receives a new consecutive part of a noisy sequence at each time instance. Each mini-column with HTM neurons models a competitive process in which neurons in the predictive state emit spikes sooner than inactive neurons. HTM then deploys fast local inhibition to prevent the inactive neurons from firing, biasing the network toward the predictions. The permanence values are then updated before the process repeats at the next time instance.

In short, a cyclic sequence of activations, leading to predictions, followed by activations again, forms the basis of HTM's sequence memory. HTM continually learns sequences with structure by verifying its predictions. When the structure changes, the memory forgets the old structure and learns the new one. Since the system learns by confirming predictions, it does not require any explicit teacher labels. HTM keeps track of multiple candidate sequences with common subsequences until further input identifies a single sequence. The use of SDRs makes networks robust to noisy input, natural variations, and neuron failures [35, 37].

5.5 Reference frames

We have outlined how networks of mini-columns learn predictive models of changing sequences. Here, we start to address how HTM learns predictive models of static objects, where the input changes due to sensor movements. We consider how HTM uses allocentric reference frames, i.e., frames around objects [24, 39]. Sensory features are related to locations in this object-centrix reference frame. Changes due to movement are then mapped to changes of locations in the frame, enabling predictions of what sensations to expect when sensors move over an object.

To understand the distinction between typical deep learning representations and a reference frame-based representation, consider JPEG images of a coffee mug vs. a 3D CAD model of the mug. On the one hand, an AI system that uses just images would need to store hundreds of pictures taken at every possible orientation and distance to make detailed predictions using image-based representations. It would need to memorize the impact of movements and other changes for each object separately. Deep learning systems today use such brute-force image-based representations.

On the other hand, an AI system that uses a 3D CAD representation would make detailed predictions once it has inferred the orientation and distance. It only needs to learn the impact of movements once, which applies to all reference frames. Such a system could then efficiently predict what would happen when a machine's sensor, such as an artificial fingertip, moves from one point on the mug

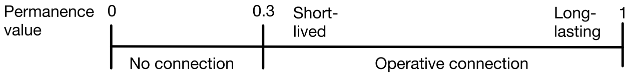

Fig. 9 The permanence value's effect on connections between HTM neurons

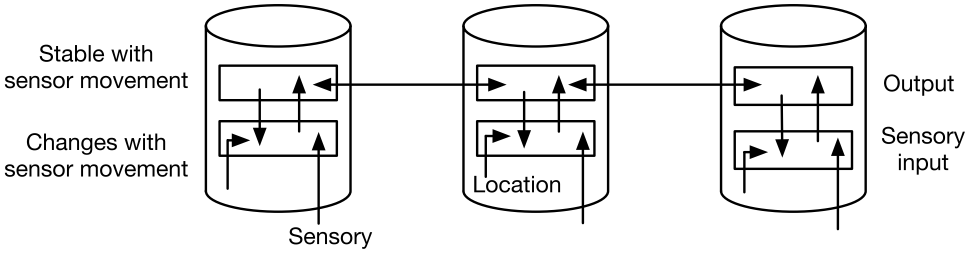

Fig. 10 Three connected cortical columns in a region. The layer generating the location signal is not shown. Adapted from [42]

to another, independent of how the cup is oriented relative to the machine.

Every cortical column in HTM maintains allocentric reference frame models of the objects it senses. Sparse vectors represent locations in the reference frames. A network of HTM neurons uses these sparse location signals as context vectors to make detailed sensory predictions (Fig. 8c). Movements lead to changes in the location signal, which in turn leads to new forecasts.

HTM generates the location signal by modeling *grid cells*, initially found in the entorhinal cortex, the primary interface between the hippocampus and neocortex [40]. Animals use grid cells for navigation and represent their body location in an allocentric reference frame, namely that of the external environment. As an animal moves around, the cells use internal motion signals to update the location signal. HTM proposes that every cortical column in the neocortex contains cells analogous to entorhinal grid cells. Instead of representing the body's location in the reference frame of the environment, these cortical grid cells represent a sensor's location in the object's reference frame. The activity of grid cells represents a sparse location signal used to predict sensory input (Fig. 8c).

5.6 Sensorimotor integration

Sensorimotor integration occurs in every cortical column of the neocortex. Each cortical column receives sensory input and sends out motor commands [41]. In HTM, sensorimotor integration allows every cortical column to build reference-frame models of objects [24, 39, 42]. Each cortical column in HTM contains two layers called the sensory input layer and the output layer, as shown in Fig. 10. The sensory input layer receives direct sensory input and contains mini-columns of HTM neurons, while the output layer contains HTM neurons that represent the sensed object. The sensory input layer learns specific object feature/location combinations, while the output layer learns representations corresponding to objects (see [42] for details).

During inference, the sensory input layer of each cortical column in Fig. 10 receives two sparse vector signals. First, a location signal computed by grid cells in the lower half of the cortical column (not shown) [24, 30, 39, 43]. Second, feedforward sensory input from a unique sensor array, such as the area of an artificial retina. The input layer combines sensory input and location input to form sparse representations that correspond to features at specific locations on the object. Thus, the cortical column knows both what features it senses and where the sensor is on the object.

The output layer receives feedforward inputs from the sensory input layer and converges to a stable pattern representing the object. The second layer achieves convergence in two ways: (1) by integrating information over time as the sensors move relative to the object and (2) spatially via lateral (sideways) connections between columns that simultaneously sense different locations on the same object. The lateral connections across the output layers permit HTM to quickly resolve ambiguity and deduce objects based on adjacent columns' partial knowledge. Finally, feedback from the output layer to the sensory input layer allows the input layer to more precisely predict what feature will be present after the sensor's subsequent movement.

The resulting object models are stable and invariant to a sensor's position relative to the object, or equivalently, the object's position relative to the sensor. All cortical columns in any region of HTM, even columns in the low-level regions (Fig. 4), can learn representations of complete objects through sensors' movement. Simulations show that a single column can learn to recognize hundreds of 3D objects, with each object containing tens of features [42]. The invariant models enable HTM to learn with very few examples since the system does not need to sense every object in every possible configuration.

5.7 The need for cortical hierarchies

Because the spatial extent of the lateral connections between a region's columnar output layers limits the ability to learn expansive objects, HTM uses hierarchies of regions to represent large objects or combine information from multiple senses. To illustrate, when a person sees and touches a boat, many cortical columns in the visual and somatosensory hierarchies, as illustrated in Fig. 4, observe different parts of the boat simultaneously. All cortical columns in each of the two hierarchies learn models of the boat by integrating over movements of sensors. Due to the non-hierarchical connections, represented by the horizontal connections in Fig. 4, inference occurs with the movement of the different types of sensors, leading to a rapid model building. Observe that the boat models in each cortical column are different because they receive different information depending on their location in the hierarchy, information processing in earlier regions, and signals on lateral connections.

Whenever HTM learns a new object, a type of neuron, called a displacement cell, enables HTM to represent the object as a composition of previously learned objects [24]. For example, a coffee cup is a composition of a cylinder and a handle arranged in a particular way. This object compositionality is fundamental because it allows HTM to learn new physical and abstract objects efficiently without continually learning from scratch. Many objects exhibit behaviors. HTM discovers an object's behavior by learning the sequence of movements tracked by displacement cells. Note that the resulting behavioral models are not, first and foremost, internal representations of an external world but rather tools used by the neocortex to predict and experience the world.

5.8 Toward a common cortical algorithm

Cortical columns in the neocortex, regardless of their sensory modality or position in a hierarchy, contain almost identical biological circuitry and perform the same basic set of functions. The circuitry in all layers of a column defines the *common cortical algorithm*. Although the complete common cortical algorithm is unknown, the current version of HTM models fundamental parts of this algorithm.

Section 5.4 describes how a single layer in a cortical column learns temporal sequences. Different layers use this learning technique with varying amounts of neurons and different learning parameters. Section 5.5 introduces reference frames, and Sect. 5.6 outlines how a cortical column uses reference frames. Finally, Sect. 5.7 describes how columns cooperate to infer quickly and create composite models of objects by combining previously learned models. A reader wanting more details about the common cortical algorithm's data formats, architecture, and learning techniques should study references [24, 30, 33–37, 39, 42, 44, 45].

Although HTM does not provide a complete description of the common cortical algorithm, we can summarize its novel ideas, focusing on how grid and displacement cells allow the algorithm to create predictive models of the world. The new aspects of the model building are [24, 30]:

- Cortical grid cells provide every cortical column with a location signal needed to build object models.
- Since every cortical column can learn complete models of objects, an HTM system creates thousands of models simultaneously.
- A new class of neurons, called displacement cells, enables HTM to learn how objects are composed of other objects.
- HTM learns the behavior of an object by learning the sequence of movements tracked by displacement cells.
- Since all cortical columns run the same algorithm, HTM learns conceptual ideas the same way it learns physical objects by creating reference frames.

Each cortical column in HTM builds models independently as if it is a brain in itself. The name *Thousand Brains Theory* denotes the many independent models of an object at all levels of the region hierarchies, and the extensive integration between the columns [24, 30].

6 Validation and future work

We have developed an understanding of the *Thousand Brains Theory* and the HTM model of the neocortex. This section discusses HTM simulation results, commercial use of HTM, a technology demonstration concerning SDRs, and the need to reverse engineer the thalamus to obtain a more complete neocortical model.

6.1 Simulation results

From Sect. 5.4, a layer in a cortical column contains an HTM network with mini-columns of HTM neurons. The fundamental properties of HTM networks are continuous online learning, incorporation of contextual information, multiple simultaneous predictions, local learning rules, and robustness to noise, loss of neurons, and natural variation in input. The Numenta research team has published simulation results [35, 37, 45] verifying these properties.

The simulations show that HTM networks function even when nearly 40% of the HTM neurons are disabled. It follows that individual HTM neurons are not crucial to network performance. The same is true for biological neurons. Furthermore, HTM networks have a cyclic sequence of operations: activations, leading to predictions, followed by activations again. Since HTM learns continuously the cycle never stops. Simulations show that with a continuing stream of sensory inputs, learning converges to a stable representation when the input sequence is stable, and changes whenever the input sequences change. It then converges to a stable representation again if there are no further changes. Hence, HTM networks continuously adapt to changes in input sequences, as the networks forget old sequence structures and learn new ones.[1] Finally, the HTM model achieves comparable accuracy to other state-of-the-art sequence learning algorithms, including statistical methods, feedforward neural networks, and recurrent neural networks [45].

Sections 5.5–5.7 outline how cortical columns learn models of objects using grid cells, displacement cells, and reference frames. Numenta's simulation results show that individual cortical columns learn hundreds of objects [39, 42, 44]. Recall that a cortical column learns features of objects. Since multiple objects can share features, a single sensation is not always enough to identify an object unambiguously. Simulations show that multiple columns working together reduce the number of sensations needed to recognize an object. The HTM reliably converges on a unique object identification as long as the model parameters have reasonable values [39].

HTM's ability to learn object models is vital to achieve general AI because cortical columns that learn a language or do math also use grid cells, displacement cells, and reference frames. A future version of HTM should model abstract reasoning in the neocortex. In particular, it should model how the neocortex understands and speaks a language, allowing an intelligent machine to communicate with humans and explain how it reached a goal.

6.2 Commercial use of HTM

Numenta has three commercial partners using HTM. The first partner, Cortical.io, uses core principles of SDRs in its platform for semantic language understanding. For example, the platform automates the extraction of essential information from contracts and legal documents. The company has Fortune 500 customers. Grok, the second partner, uses HTM anomaly detection in IT systems to detect problems early. Operators can then drill down to severe anomalies and take action before problems worsen. The third partner, Intelletic Trading Systems, is a fintech startup. The company has developed a platform for autonomous trading of futures and other financial assets using HTM. The described commercial usage documents HTM's practical relevance.

6.3 Sparsity accelerates deep learning networks

Augmented deep learning networks like AlphaZero have achieved spectacular results. Still, they are hitting bottlenecks as researchers attempt to add more compute power and training data to tackle ever-more complex problems. During training, the networks' data processing consumes vast amounts of power, sometimes costing more than a million dollars, limiting scalability. The neocortex is much more energy efficient, requiring less than 20 Watts to operate. The primary reasons for the neocortex's power efficiency are SDRs and the highly sparse connectivity between the neocortical neurons. Deep learning has dense data representations and dense networks.

Numenta has created sparse deep learning networks on field programmable gate array (FPGA) chips requiring no more than 225 Watts to process SDRs. Using the Google Speech Commands dataset, 20 identical sparse networks running in parallel performed inference 112 times faster than multiple comparable dense networks running in parallel. All networks ran on the same FPGA chip and achieved similar accuracy (see [46]). The vast speed improvement enables massive energy savings, the use of much larger networks, or multiple copies of the same network running in parallel. Perhaps, most importantly, sparse networks can run on limited edge platforms where dense networks do not fit.

6.4 Thalamus

The thalamus is a paired, walnut-shaped structure in the center of the brain. It consists of dozens of groups of cells, called nuclei. The neuroanatomists Ray Guillery and Murray Sherman wrote the book *Functional Connections of Cortical Areas: A New View from the Thalamus* [41] describing the tight couplings between the nuclei in the thalamus and regions of the neocortex. As shown in Fig. 11, nearly all sensory information coming into the neocortex flows through the thalamus. A cortical region sends information directly to another region, but also indirectly via the thalamus. These indirect routes between the regions strongly indicate that it is necessary to understand the operations of the thalamus to fully understand the neocortex. Guillery and Sherman make a convincing case that the computational aspects of the thalamus must be reverse-engineered and added to HTM [47].

7 Takeaways

According to this paper's biological reasoning, the improvement of existing narrow AI techniques alone is unlikely to bring about general AI. Current AI is reactive

[1] Note that the predicted state is internal to the neuron and does not drive subsequent activity. As such, for each input the network activity settles immediately until the next input arrives, and the network cannot diverge in the classic sense.

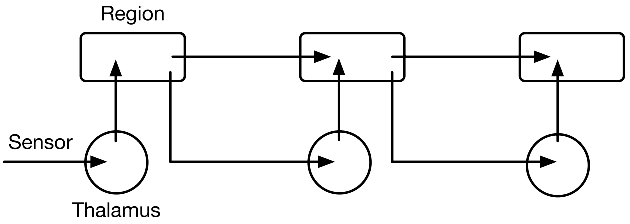

Fig. 11 The connections between the regions in the neocortex are both direct and via the thalamic nuclei. Adapted from [41, 47]

Table 1 Problems with current AI and solutions proposed by neuroscience studies of the neocortex

Existing AI	Neuroscience
Offline, batch-oriented learning	Online, continuous learning
Supervised learning	Unsupervised learning
Massive training sets	Streaming of sensory data
Fragility to noise and faults	SDRs provide robustness to noise and faults
Manual process to create system	Automatic, general system
Inefficient power usage	Highly power efficient
Symbol manipulation	Embodied reasoning*

*More research is needed

rather than proactive. While humans learn continuously from data streams and make updated predictions, today's AI systems cannot do the same. As an absolute minimum to create general AI, more biologically plausible neuron models must replace the point neurons used in artificial neural networks. The new neuron models must allow the networks to predict their future states.

Furthermore, it is necessary to use SDRs to enable multiple simultaneous predictions and achieve robustness to noise and natural variations in the data. There is a great need for improved unsupervised learning using reference frames. Finally, a particularly severe problem with current AI is the lack of sensorimotor integration providing the ability to learn by interacting with the environment.

Table 1 summarizes the actionable takeaways from the article. The left column lists fundamental problems with narrow AI, while the right column contains solutions proposed by neuroscience studies of the neocortex. HTM models these solutions. To achieve biologically constrained, human-like intelligence, AI researchers need to work with neuroscientists to understand the computational aspects of the neocortex and determine how to implement them in AI systems. They also need to reverse engineer the thalamus to understand how the neocortex's regions work together. This accomplishment would lead to an overall understanding of the neocortex and, most likely, facilitate the creation of general AI.

Numenta has published simulations of core HTM properties, and commercial products use HTM. However, there is still a need to demonstrate HTM's full potential because—in contrast to deep learning with AlphaZero—there is no HTM application with proven superhuman capabilities. A powerful application based on the neocortex would validate the importance of biologically constrained AI in general and HTM in particular. It would also convince more talented young researchers to follow the biological path toward general AI.

Funding The authors were financed by their employers.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Husain A (2017) The sentient machine: the coming age of artificial intelligence, 1st edn. Scribner, New York
2. Sejnowski TJ (2018) The deep learning revolution, 1st edn. The MIT Press, Cambridge
3. Sterling P, Laughlin S (2015) Principles of neural design, 1st edn. The MIT Press, Cambridge
4. Ballard DH (2015) Brain computation as hierarchical abstraction, 1st edn. The MIT Press, Cambridge
5. Lakoff G, Johnson M (1999) Philosophy in the flesh: the embodied mind and its challenge to western thought, 1st edn. Basic Books, New York
6. Johnson M (2017) Embodied mind, meaning, and reason: how our bodies give rise to understanding, 1st edn. The University of Chicago Press, Chicago
7. Russell S, Norvig P (2020) Artificial intelligence: a modern approach, 4th edn. Pearson, London
8. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. The MIT Press, Cambridge

9. Liebowitz J (ed) (1997) The handbook of applied expert systems, 1st edn. CRC Press, Boca Raton
10. Tesauro G (1994) TD-Gammon, a self-teaching backgammon program, achieves master-level play. Neural Comput 6:215–219
11. Minsky M (2006) The emotion machine: commonsense thinking, artificial intelligence, and the future of the human mind, 1st edn. Simon & Schuster, New York
12. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, Lillicrap T, Simonyan K, Hassabis D (2018) A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362:1140–1144
13. McAfee A, Brynjolfsson E (2017) Machine, platform, crowd: harnessing our digital future, 1st edn. W. W. Norton & Company, New York
14. Marcus G (2018) Innateness, AlphaZero, and artificial intelligence. arXiv:1801.05667
15. Marcus G (2018) Deep learning: a critical appraisal. arXiv:1801.00631
16. Hole KJ, Ahmad S (2019) Biologically driven artificial intelligence. IEEE Comput 52:72–75
17. Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, Prakash A, Kohno T, Song D (2018) Robust physical-world attacks on deep learning models. arXiv:1707.08945
18. Athalye A, Engstrom L, Ilyas A, Kwok K (2018) Synthesizing robust adversarial examples. arXiv:1707.07397
19. Broussard M (2018) Artificial unintelligence: how computers misunderstand the world, 1st edn. The MIT Press, Cambridge
20. Brooks R (2017) The seven deadly sins of AI predictions. MIT Technology Review. Oct. 6, 2017. https://www.technologyreview.com/s/609048/the-seven-deadly-sins-of-ai-predictions/
21. Hassabis D, Kumaran D, Summerfield C, Botvinick M (2017) Neuroscience-inspired artificial intelligence. Neuron 95:245–258
22. Churchland PS, Sejnowski TJ (1992) The computational brain, 1st edn. A Bradford Book, Cambridge
23. Alexashenko S (2017) Cortical circuitry. 1st ed. Cortical Productions
24. Hawkins J, Lewis M, Klukas M, Purdy S, Ahmad S (2019) A framework for intelligence and cortical function based on grid cells in the neocortex. Front Neural Circuits 12:121
25. Hawkins J, Blakeslee S (2004) On intelligence, 1st edn. Times Books, New York
26. Hawkins J (2021) A thousand brains: a new theory of intelligence, 1st edn. Basic Books, New York
27. Hawkins J (2017) What intelligent machines need to learn from the neocortex. IEEE Spectr 54:34–71
28. Costandi M (2016) Neuroplasticity, 1st edn. The MIT Press, Cambridge
29. Mountcastle VB (1997) The columnar organization of the neocortex. Brain 120:701–722
30. Numenta. A framework for intelligence and cortical function based on grid cells in the neocortex—a companion paper. 2018. https://numenta.com/neuroscience-research/research-publications/papers/thousand-brains-theory-of-intelligence-companion-paper/
31. Spruston N (2008) Pyramidal neurons: dendritic structure and synaptic integration. Nat Rev Neurosci 9:206–221
32. Buxhoeveden DP, Casanova MF (2002) The minicolumn hypothesis in neuroscience. Brain 125:935–951
33. Ahmad S, Hawkins J (2015) Properties of sparse distributed representations and their application to Hierarchical Temporal Memory. arXiv:1503.07469
34. Ahmad S, Hawkins J (2016) How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites. arXiv:1601.00720
35. Cui Y, Ahmad S, Hawkins J (2017) The HTM spatial pooler-a neocortical algorithm for online sparse distributed coding. Front Comput Neurosci 11:111
36. Purdy S (2016) Encoding data for HTM systems. arXiv:1602.05925
37. Hawkins J, Ahmad S (2016) Why neurons have thousands of synapses, a theory of sequence memory in neocortex. Front Neural Circuits 10:23
38. Major G, Larkum ME, Schiller J (2013) Active properties of neocortical pyramidal neuron dendrites. Annu Rev Neurosci 36:1–24
39. Lewis M, Purdy S, Ahmad S, Hawkins J (2019) Locations in the neocortex: a theory of sensorimotor object recognition using cortical grid cells. Front Neural Circuits 13:22
40. Moser EI, Kropff E, Moser M-B (2008) Place cells, grid cells, and the brain's spatial representation system. Annu Rev Neurosci 31:69–89
41. Sherman SM, Guillery RW (2013) Functional connections of cortical areas: a new view from the thalamus, 1st edn. The MIT Press, Cambridge
42. Hawkins J, Ahmad S, Cui Y (2017) A theory of how columns in the neocortex enable learning the structure of the world. Front Neural Circuits 11:81
43. Rowland DC, Roudi Y, Moser M-B, Moser EI (2016) Ten years of grid cells. Annu Rev Neurosci 39:19–40
44. Ahmad S, Hawkins J (2017) Untangling sequences: behavior vs. external causes. https://www.biorxiv.org/content/early/2017/09/19/190678
45. Cui Y, Ahmad S, Hawkins J (2016) Continuous online sequence learning with an unsupervised neural network model. Neural Comput 28:2474–2504
46. Numenta. Sparsity enables 100x performance acceleration in deep learning networks: a technology demonstration. Numenta Whitepaper, version 2 2021
47. Hawkins J (2018) Two paths diverged in the brain, Ray Guillery chose the one less studied. Eur J Neurosci 49:1005–1007