20 Million-Dollar Problems for Any Brain Models and a Holistic Solution: Conscious Learning

Juyang Weng[1,2]

[1]Brain-Mind Institute
[2]GENISAMA, Okemos, MI 48864 USA

Abstract*—This is a theoretical paper. It raises 20 open problems each of which is estimated to require one million dollars of investment or more. They are (1) the image annotation problem (e.g., retina is without bounding box to learn, unlike ImageNet), (2) the sensorimotor recurrence problem (e.g., all big data sets are invalid), (3) the motor-supervision problem (e.g., impractical to supervise motors throughout lifetime), (4) the sensor calibration problem (e.g., a life calibrates the eyes automatically), (5) the inverse kinematics problem (e.g., a life calibrates all redundant limbs automatically), (6) the government-free problem (i.e., no task-aware homunculus inside a brain), (7) the closed-skull problem (e.g., supervising hidden neurons is biologically implausible), (8) the nonlinear controller problem (e.g., a brain is a nonlinear controller but task-nonspecific), (9) the curse of dimensionality problem (e.g., a set of global features is insufficient for a life), (10) the under-sample problem (i.e., few available examples in a life), (11) the distributed vs. local representations problem (i.e., how both representations emerge), (12) the symbol problem (also called grounding problem, thus must be free from any symbols), (13) the local minima problem (so, avoid error-backprop learning and Post-Selections), (14) the abstraction problem (i.e., require various invariances and transfers), (15) the compositionality problem (e.g., metonymy beyond those composable from sentences), (16) the smooth representations problem (e.g., brain representations are globally smooth), (17) the motivation problem (e.g., including reinforcements and various emotions), (18) the global optimality problem (e.g., avoid catastrophic memory loss and Post-Selections), (19) the auto-programming for general purposes (APFGP) problem, (20) the brain-thinking problem. The paper discusses also why the proposed holistic solution of conscious learning [1], [2] solves each.

I. Introduction

Many experts have stated that a human brain is probably the most complex objects ever-existed [3], [4]. The Third World Science and Technology Development Forum (WSTDF) co-sponsored by UNESCO published "Top 10 Scientific Issues Concerning Human Social Development 2021" [5]. In the Information Field, the first issue is: "How does the human brain process information and how do humans form intelligence?"

A. Blind men and an elephant

The issue of understanding a brain is like the well-known parable called "blind men and an elephant", shown in Fig. 1. The nature—a brain in this case—exists first. It is humans who partitioned the science of nature into multiple disciplines of limited scopes, such as biology, neuroscience, psychology, electrical engineering, computer science, mathematics, and physics.

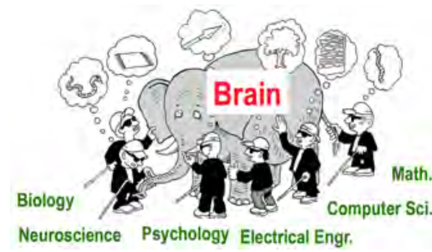

Fig. 1. The problem of understanding a brain is like "blind men and an elephant", especially "neuromorphic". Adapted from Hans Moller, mollers.dk.

On one hand, this partition is helpful for humans to organize curricula in universities and specifying job categories. Courses and degrees in a discipline make the scope of learning manageable and job scope better understood.

On the other hand, such a partition of disciplines has created artificial boundaries about nature that does not seem to have such boundaries originally. In the case of understanding the brain, each discipline is like a blind man. It focuses on a limited subject scope, which is insufficient for understanding how the brain works.

B. Three AI schools

Understandably, facing tremendous challenges, Artificial Intelligence (AI) has a limited scope in modeling natural intelligence. There are three major schools [6] in AI, symbolic, connectionist and developmental.

Although all the three schools aim to simulate natural intelligence, their emphases are different. The symbolic school uses symbols as basic elements in the *representations* of an AI system. The connectionist school uses neurons as basic *processing elements* in an AI system. The developmental school stresses *task-nonspecificity* [7] during the development throughout lifetime.

Because the symbols are about *representations* and neurons are about *computations*, the symbolic school and the connectionist schools have been widely divided ever since the early years of AI. The developmental school provides a platform on which the symbolic school and the connectionist school can *unify*. Both schools can contribute to task-nonspecific development.

In Proc. International Joint Conference on Neural Networks, pp. 1-9, Padua, Italy, July 18-23, 2022.

C. Symbolic school

A symbol for an AI system is defined by a human programmer along with its meaning. For example, "a symbol H represents a human". Such a definition is in a design document of the AI system, not read and understood by the AI system.

A human who handcrafted the AI program may link an ASCII code of a symbol "H" to a text string "human". Thus, the AI system does not understand any symbols like "H" and "human" that it uses. All the machine does is to follow the programed rules the human handcrafted in terms of human defined symbols (e.g., in a C language).

In the symbolic AI system, processors are separate from the symbols. A large system of the symbolic school may require many symbols, such as CYC, WordNet and EDR [8], [9].

Some scholars thought that symbols are "neat" and "logical" [10]. This is because they define symbols and therefore they understand their own design, but the machines do not understand their design. That is the root reasons that symbolic AI systems are brittle. Symbolic computation is insufficient.

D. Connectionist school

An artificial neuron is only a processor, often not an element of representation. Therefore, there is always an issue of representation with a neural network. This is indeed one of the most challenging problems in the connectionist school. For this reason, neural networks have been criticized to be scruffy [10] and do not abstract well [11]. Convolutional Neural Networks (CNNs) [12]–[16] and LSTM [17] are recently called deep learning networks. But representations in CNNs are flawed.

Many of them were trained using error-backprop, which only greedily fits three data sets, training set, verification set, and test set. In what is later called Extreme Learning Machine (ELM) [18], hidden features are random and they do not change at all during training. Some of them have also become increasingly large without a limit on memory resource and the search for the luckiest network is speeded up using Graphic Processing Units (GPUs).

Weng 2021 [19] argued that both the symbolic school and the connectionist school suffer from Post-Selection—selection of systems using test sets as the programmer possesses the test sets (typically the test error and verification error are both small). In Krizhevsky & Hinton 2017 [16] the test error is even smaller than the verification error.

In the symbolic school, the human programmer post-selects symbolic meanings and other symbolic representations since they possess the test set originally. This is like exams have been leaked.

In the connections school, the human programmer train multiple systems, each starting from a different random seed for weights. A search of trained networks is performed on random seeds and many hyper-parameters, using test sets as the programmer possesses the test sets. The test set was "unseen" by the luckiest system prior to the test, but the Post-Selection of the luckiest system saw the test set.

Later in Section XIV, I will present a theorem that states that without a limit on the memory size, a special nearest-neighbor classifier produces any nonzero verification error and nonzero test error, as long at the programmer possess three data sets—the training set, the verification set and the test set. It is almost always true that the authors of a paper possess the three data sets.

Consider a deployment of AI systems to users. A user must buy a single system that works on his tasks and data. The seller does not possess user's data. The user cannot risk failures by purchasing many systems and then trying each of them.

Both the symbolic school and the connectionist school so far have failed to produce a single system that performs well on programmer-unpossessed tasks and data.

One may say that during computer game competitions, the programmer does not possess the test set. Weng 2021 [19] argued that in such competitions there is evidence where humans were probably involved in interactive and on the fly selection of multiple trained systems. If this is true, the competition is between multiple humans with computer assistance and a single human without any computer assistance.

E. Developmental school

The developmental school, proposed in Weng et al. [7], emphasizes task-nonspecificity across the entire life. A developmental program aims to simulate the functions of genes from two parents for the lifelong learning of the child. It does not simulate the genes themselves.

Because a life takes an open array of potential tasks, from simple to complex, any symbols seem to be inappropriate. Section XIII discusses this symbol issue. This issue is controversial since even connectionist school has used symbols often (e.g., as class labels).

Does a brain have any element that corresponds to a symbol? For example, my parents never used computers. It is unlikely that their genes have mechanisms that are specific to symbols that I learned in computer science and many other disciplines. The brain allows me to pronounce "C language", but "C" corresponds to many motor neurons that fire in my way. These neurons are involved in pronouncing other sounds. Thus, none of these motor neurons correspond to a symbol in a one-to-one correspondence like a symbol requires.

Furthermore, since learning in a life must be carried out on the fly in an incremental way, developmental learning must be incremental, as a brain does. These considerations raise some fundamental problems discussed in this paper, such as the optimization problem.

Since a human brain takes tests at different lifetimes, errors made by a learner must be recorded across the entire life. The developmental program must work well for each life, without a possibility of Post-Selection.

The solution to the optimization problem at each instant spans the entire life. The Developmental School appears to be the only school that has successfully avoided the Post-Selection problem. Namely, the developmental school only trains one network for a life, and the network must be successful without any possibility of Post-Selection.

Fig. 2. An image annotation of a human face by a polygon that excludes hair, facial wears, clothes, other body parts, and other objects in a cluttered scene. Figure courtesy of [13].

Below, let us discuss 20 of million dollar problems, one problem in a section, taking a total of 20 sections. The order of the problems was chosen to be from simple to complex. The solution in [1], [2] to all 20 problems below is holistic since all the solutions are from a single network called Developmental Network 2 (DN-2) [20], [21]. Finally, Section XXII provides concluding remarks. Let us tolerate the conceptual proofs below because the nature of the problems is conceptual.

II. The image annotation problem

Definition 1 (Image annotation problem): During machine training, for each image frame a learning algorithm requires the annotation of a particular region that encloses the projection of the object to be attended to and excludes other elements in the cluttered scene.

Fig. 2 gives an example, from the work that seems to be the first deep neural network for 3D. Such annotations have been widely used later, such as rectangles in ImageNet [22].

In contrast, the retina is without a bounding box to mark something to learn. Solution: motor neurons automatically direct attention on retina.

The solution allows the learner to learn on the fly, denoted by a mathematical formulation of a neural network [23]. Namely, a neural network self-generates its state/action for each time frame, in the motor area. Suppose X is the space of input images, Y the space of the hidden areas (as features), and Z the space of the motor area. This mathematical formulation unfolds time $t = 0, 1, 2, ...$ as discrete indices of real time for $X(t), Y(t), Z(t)$:

$$\begin{bmatrix} Z(0) \\ Y(0) \\ X(0) \end{bmatrix} \xrightarrow{f_0} \begin{bmatrix} Z(1) \\ Y(1) \\ X(1) \end{bmatrix} \xrightarrow{f_1} \begin{bmatrix} Z(2) \\ Y(2) \\ X(2) \end{bmatrix} \xrightarrow{f_2} ... \quad (1)$$

where $\rightarrow$ means each neuron on the right side adaptively links from some or all neurons on the left side. Such a link $\rightarrow$ is established automatically, incrementally, and adaptively as a multivariate mapping

$$f_t : (X(t), Y(t), Z(t)) \mapsto (X(t+1), Y(t+1), Z(t+1)) \quad (2)$$

using (unsupervised) Hebbian learning [24], i.e., the dually optimal Lobe Component Analyses (LCA) [25].

For example, the state/action $Z(t)$ affects which neurons $\{n\}$ in $Y(t+1)$ to win the competition and to fire. The receptive field in $X(t)$ of $\{n\}$ corresponds to the attended *region* and *pattern*) in $X(t)$ (like the right pane in Fig. 2) without a need for manual annotation of the receptive field. The attended *pattern* in $X(t)$, typically at many retina locations, is important for (attention directed) object detection. Thus, we have proven that object recognition and object detection are unified by attention:

Theorem 1 (Auto-attention): The image annotation problem is solved by an automatic attention scheme that unifies object recognition and object detection (also segmentation).

Networks that are hierarchically feedforward in nature, such as CNN [12]–[16], LSTM [17] and ELM [18] cannot conduct attention since attention should be purpose-based and, therefore, top-down in nature. Top-down (e.g., Z-to-Y) and bottom-up (e.g., X-to-Y and Y-to-Z) information needs to be highly integrated as in Eq. (1) across such a hierarchy. This holistic solution [20], [21] is further discussed in Sec. XII.

III. The sensorimotor recurrence problem

The following definition is from Eq. (1).

Definition 2 (Sensorimotor recurrence): At each time t the action of the learner affects what sensory image is received at time $t + 1$.

Theorem 2 (Big-data flaw): Any batch data sets are flawed because of their violations of the sensorimotor recurrence.

Proof: The proof follows directly from Definition 2 because such batch data sets are collected without taking into account learner's actions. ■

Namely all batch data sets disallow the learner to learn the causal relationship between what it does and what it senses.

Unlike "big data" [19], the next sensory input is altered by the current action. Solution: on-the-fly conscious learning in DN-2 avoids static "big data".

Theorem 3 (On the fly learning): Grounded and on the fly learning satisfies sensorimotor recurrence.

Proof: "Grounded" means that the actions at time t from the learner apply to the real world and the sensors of the learner receive the sensory effects at $t+1$ of the actions from the real world. "On the fly" means that the sensor gets effects with a negligible delay from the grabbed sensory frame at $t + 1$. Therefore, Definition 2 is satisfied. ■

On the fly learning depends on the required response time, e.g., 100Hz for autonomous driving. Assuming a learner can act, so called "active learning" is batch, not on-the-fly.

IV. The motor-supervision problem

Definition 3 (Motor-supervision): A motor supervision means that a teacher imposes the corresponding effectors of the learner in real time. This is impossible if the effectors are not imposable (e.g., vocal tract motor or attention motor).

Unlike motor-imposed learning, motors of a child are mostly of free will, such as arms, legs, vocal tract and various attention. Solution: motor-unsupervised learning like a biological life. This solution has not yet been demonstrated on real robots but has been demonstrated by animals.

Theorem 4 (Motor-unsupervised): Motor-unsupervised learning solves the motor-supervision problem.

Proof: During motor-unsupervised learning, the teacher does not directly impose learner's motor. ■

Practical learning modes without motor-imposition include (1) reinforcement learning where reinforcers are available in a time-sparse way and may be time delayed, and (2) communicative learning through conversations, demonstrations, and classroom teaching. Weng proposed conscious learning [1], [2], which avoids motor supervision. He suggested teaching innate behaviors before birth that may facilitate generation of earlier imitation behaviors without a need for any motor impositions from teachers, but not of special-purpose.

V. The sensor calibration problem

At the birth time of a network, sensors are connected with the network but they have not been calibrated.

Definition 4 (Sensor calibration): The problem of sensor calibration is to establish a mapping between the sensing elements in a sensor and the learner's actions so that the effect of the actions matches the sensed image.

This definition is free of a sensor model, thus, is more general than a typical engineering method for camera calibration where one only needs to estimate the focal length and compensate undesirable effects such as lens distortions.

Unlike traditional camera calibration, a life calibrates its eyes autonomously. Solution: a network (e.g., DN-2) calibrates all sensors through autonomous experience as trial and error. This solution has not yet been demonstrated on real robots but an algorithm has been published recently [1], [2] motivated by biological solutions demonstrated by animals.

Definition 5 (Conscious learning): Consciousness is defined [26] as (1) the state of being aware of a) something within oneself, b) external object state or fact, and c) social or political cause; (2) the state of being characterized by sensation, emotion, volition, and thought; (3) the totality of conscious state of an individual; (4) the normal state of conscious life; and (5) the upper level of mental life. Definition 9 below does not rule out any of these levels (1) to (5). By conscious learning [1], [2], the learner must be conscious of above at each instant t so as to be more conscious of "*longer and higher*" context of above at instant $t+1$.

This definition requires a learner to be aware of its goal, to judge whether the current action reaches its goal and to generate the next actions to try. This kind of consciousness should be basic components of conscious learning.

Due to the task-nonspecificity of the developmental school, the "height" of goals and next actions becomes higher and higher automatically throughout lifetime experience.

Theorem 5 (Sensor calibration): If handcrafting machine consciousness is impractical, autonomous sensor calibration requires conscious learning.

Proof: We prove by examples. Suppose handcrafting machine consciousness is impractical. We need to prove that autonomous sensor calibration requires consciousness. Suppose that a sensor is not yet calibrated. The network needs to learn visual attention, object recognition, and object detection. To learn the relation between a motor and an object sensed, the network needs to generate a purpose, such as interaction between the motor and the object. Reaching of the motor using arm and the object is an example. The learning needs to be conscious whether the purpose is being executed, whether the goal of the purpose is getting closer or farther, and what is the next action to get closer, and whether the goal has been reached. All the concepts involved are at levels 1 and 2 in Definition 5. Therefore, the autonomous sensor calibration requires conscious learning, at least at levels 1 and 2. ■

Imprinting [27], present in early life, is a good example of conscious learning. A duckling automatically tries actions while calibrating its sensors so that the goal of its emergent behaviors (follow a moving, early sensed familiar object) look like following the mother.

Calibration of sensors here and calibration of motors in the next section are carried out concurrently by the learner. The holistic solution is task-nonspecific, different from [28].

VI. The inverse kinematics problem

If a human is deaf at birth, he has a severe deficit in spoken language [29]. This is a piece of evidence for autonomous calibration of vocal tract motor through trial and error since hearing and goals are necessary in calibration of vocal tract.

Definition 6 (Inverse kinematics): Forward kinematics means computation from joint angles of a robot arm (or a limb of a life) to the position and orientation of the end effector. Inverse kinematics is an inverse process—from the position and orientation of the end effector compute the required joint angles.

In general the inverse problem involves not only kinematics, but also dynamics, such as force and weight.

The solution of inverse kinematics is infinitely many if a robot arm is redundant—having more than 6 degrees of freedom. Human arms and legs are redundant. Like sensory calibration, a biological life seems to automatically calibrate its redundant limbs via trial and error.

Unlike solving an engineering solution for redundant arms, a life calibrates its limbs autonomously. Solution: a network (e.g., DN-2) calibrates all effectors through trial and error. This solution has not yet been demonstrated on real robots but an algorithm has been published recently by Weng's group [1], [2] motivated by biological solutions demonstrated by animals.

Like sensory calibration, the process of motor calibration also requires the learner to be aware of its goal, to judge whether the current action reaches its goal and to generate the next actions to try. Again, these capabilities should be basic components of conscious learning.

Theorem 6 (Motor calibration): If handcrafting machine consciousness is impractical, autonomous motor calibration requires conscious learning.

Proof: The proof is analogous to the proof for Theorem 5, if we consider that the sensor calibration and motor calibration are concurrent. ■

VII. The government-free problem

Differently, Brooks [30] proposed "no central purposeful locus of control", but he avoided consciousness.

Definition 7 (Government-free): A government within a learning system is an entity that meets the following conditions conjunctively: (1) the government is task-aware, (2) the government serves a role of organization, and (3) other elements of the system obey the directives from the government. Government-free means absence of such governments inside the brain skull. Government-free is more demanding than task-nonspecificity.

This definition does not exclude a task-aware government outside the skull, since such an external government is part of the environment. The genome of a life within the skull (present in the nucleus of each cell) is not task-aware, since even a fruit fly is not task-specific. Genes only regulate the development of a learner and does not organize either.

Error backprop in a CNN has a human as government who is aware of task—shift-invariant pattern recognition. The human organizes how every neuron uses the shift-invariant property as convolution. Let us consider further details.

Case 1: If a human designs symbols within a network (e.g., LSTM) and assigns the symbols to *some individual neurons* (e.g., task-specific gates) of the network, this human is a government within the network since he is task-aware.

Case 2: If a human designs symbols within a network and assigns roles to *blocks* in a functional block diagram, e.g., [31], this human is a government within the network.

Case 3: In the symbolic AI school, a human programmer designs symbolic representations for a task that is assigned to a *computer program* or *network*. This human is a government within the symbolic AI system since he is task-aware.

All the 3 cases do not solve the government-free problem.

Unlike a task-specific program with a given goal, the brain is without a task-aware homunculus inside. Solution: each hidden neuron finds its own roles and competitors (e.g., DN-2). This solution is like market economy and avoids Cases 1 to 3.

Theorem 7 (Government-free): A conscious learning system must be government-free.

Proof: Suppose that a conscious learning system has a task-aware government inside (e.g., a human teacher), it is the human teacher that is task-aware, not the learning system. After we remove the human from the system, the remaining system is not even task-aware, let alone consciousness. ■

VIII. THE CLOSED-SKULL PROBLEM

Definition 8 (Closed-skull): A brain (or a network) is closed in its skull so that the brain is off-limit to direct manipulations by the skull-external environment. Task-nonspecific, sensor/motor driven, auto-wiring and patterning are allowed.

Note that a brain (or a network) auto-wires with its sensors and effectors, so that it can indirectly interact with the skull-external environment, but only via its sensors and effectors.

Unlike a symbolic network, the brain is not open to allow an external teacher to assign every neuron a role. Solution: hidden neurons in a network (e.g., DN-2) all learn unsupervised.

Theorem 8 (Closed-skull): If all hidden neurons inside the skull learn unsupervised, the system solves the closed-skull problem.

Proof: Since the skull is closed, direct supervision from external teachers are banned. Since all neurons inside the skull learn unsupervised, using signals from other neurons, sensors and effectors, this system solves the closed-skull problem. ■

An example of unsupervised learning is Hebb's mechanism [32], which states: "Let us assume that the persistence or repetition of a reverberatory activity (or 'trace') tends to induce lasting cellular changes that add to its stability.... When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

The LCA [33] is a dually optimal version of the above (vaguely stated) Hebb's mechanism.

IX. THE NONLINEAR CONTROLLER PROBLEM

A simple control system compares the value of the process variable (e.g., room temperature) being controlled with the desired setpoint (e.g., 20°C), and applies the difference as a control signal to bring the process variable output of the plant (e.g., room) to the same value as the setpoint.

A complex control system (e.g., an airplane) has many states, called state-based control. Under each state, there are multiple process variables. Under each combination of state and process variables, different desired setpoins are given. For example, a humanoid dance robot has a complex controller.

The Kalman filter [34] is a widely used model of state-based control. It has a plant model (including process variables) that formulates how the plant changes with time, and an observation model that formulates how measurements are produced by the plant.

If the both the plant model and the observation model can be well modeled by linear expressions (i.e., using only constant matrices), the controller is linear. Otherwise, the controller is nonlinear. Nonlinear control is a largely unsolved problem.

Like other AI areas, the area of control systems has been dominated by task-specific controls, e.g., controlling a humanoid robot to keep balance, walk, run, climb stairs, and somersault. At any time, the robot can only do one task till completion, where the task is given by a human commander. The robot is not able to decide for itself in a natural environment which task is necessary at any time, the goals, and what to do if the task fails.

Definition 9 (General-purpose control): At any time of operation, a general-purpose controller decides what task it should take up, what goals, how it executes the task, and what to do if it executes the task well, not well, or a new environmental situation occurs so the task should be changed.

Unlike nonlinear controller with a given task, the brain is without a given task. Solution: a network (e.g., DN-2) learns an open-ended list of tasks and their goals from the real world. From Definition 9, we have the following theorem.

Theorem 9 (General-purpose control): If handcrafting machine consciousness is impractical, general-purpose control requires conscious learning.

Proof: From the given condition that handcrafting machine consciousness is impractical, general-purpose control requires consciousness of what task is required, which may involved all levels 1 to 5 of consciousness. Thus, learning of consciousness (and also intelligence) is required. ■

X. The Curse of Dimensionality Problem

Definition 10 (Curse of dimensionality): The curse of dimensionality refers to various phenomena that arise for using high dimensional spaces that do not occur in low-dimensional settings.

In a low dimension, each dimension corresponds to a global feature. In a high dimension, each dimension may or may not correspond to a global feature.

Any biological brain deals with a high dimension. In a sensory area, a dimension may correspond to a cone receptor in the retina, a hair cell in the cochlea, a touch receptor in the skin. In a motor area, a dimension may correspond to a motor neuron of a muscle. In the hidden brain, each dimension may correspond to a neuron or a connection to a neuron.

Many phenomena of the curse of dimensionality problem boil down to the fact that more global features may give a worse performance, unlike a low dimensional setting. Solution: a network (e.g., DN-2) that automatically emerges a hierarchy of features, from concrete at low levels of the hierarchy to gradually more abstract at high levels of the hierarchy.

Theorem 10 (Curse of dimensionality): A hierarchy of features at multiple levels, concrete at low levels to abstract at high levels, addresses the curse of dimensionality problem. However, the hierarchy is not a cascade, but instead a complex network where a high level concept may connect to and from any low levels where the scale is small (e.g., a short edge detector is (1) linked to the motor concept "edge" that is locational invariant for edge *recognition* and (2) linked from the motor concept "edge" for edge *detection*).

Proof: To facilitate understanding, let us prove informally. Suppose the sensory area X is the most concrete and the motor area Z is the most abstract. By abstract, we mean degree of invariance. Consider four concepts, location, type, scale, and orientation. For example, the motor concept *type* of a 3D object is invariant to location, scale, and orientation. The motor concept *location* is invariant to type, scale, and orientation. In general, one concept is invariant to all three other concepts. An early neuron is concrete, specific to location, type, scale and orientation. The hidden area Y serves as a two-way fluid-like bridge between X and Z. Earlier spawn neurons establish coarse connections between X and Z, responsible for early and coarse features and behaviors (e.g., crawl). Later spawn neurons establish fine connections that refine the existing connections for later fine features and behaviors (e.g., walk). Middle neurons in the hierarchy are intermediately concrete and intermediately abstract since they have middle-size receptive fields and partially invariances. Neurons situated later in the hierarchy use earlier and middle neurons to increase the degrees of invariance and therefore are more abstract. If the motor area is also concrete, e.g., motor neurons for vocal tract that produces sounds, there is also a motor hierarchy in a reverse way. The sensory hierarchy and motor hierarchy are built in a mixed way, not separately built. ■

XI. The Under-sample Problem

Definition 11 (Under-sample): Under-sample means in a biological life, few sample events are available (e.g., examples in a lecture), much smaller than the number of pixels in an image.

Solution: a network (e.g., DN-2) learns abstract concepts in events from few samples in motor space using LCA [25] and then transfers, even when the samples are very far.

Theorem 11 (Under-sample): The feature hierarchy in Theorem 10 enables successful generalization from under-sample.

Proof: At an adult age, a mature hierarchy has been learned from earlier experience. Undersamples do not cause high novelty in an earlier area (e.g., edges are old) or in a later areas (e.g., objects are old). The novelty (i.e., new patterns) occurs mainly in the middle areas of the hierarchy. Hebbian learning enables the establishment of links between object features in the motor areas and the rule features in the motor areas from undersample. But each event typically has many temporal frames so that LCA updates are many. ■

XII. The Distributed vs. Local Representations Problem

Consider a high dimensional space $A = R^n$ with dimension $n > 1$ representing an area of n neurons. The firing pattern of A is denoted as $\mathbf{y} = (y_1, y_2, ..., y_n)$, where $y_i \geq 0$, $i = 1, 2, ..., n$, is the response of neuron i. To define the distributedness of $\mathbf{y}$, using the sum of components $s = \sum_{i=1}^{n} y_i$, so that $\mathbf{y}_p = \mathbf{y}/s = (y_1/c, y_2/c, ..., y_n/c) = (p_1, p_2, ..., p_n)$ can be considered as a distribution of discrete properties. The entropy of $\mathbf{y}$

$$H(\mathbf{y}) = -\sum_{i=1}^{n} p_i \log p_i$$

is defined as the distributedness of $\mathbf{y}$. If only one neuron fires in A, e.g., $\mathbf{y} = (1, 0, 0, ..., 0)$, the distributedness reaches the minimum $H(\mathbf{y}) = 0$. If all neurons fire at the same value, $\mathbf{y} = (1, 1, ..., 1)$, the distributedness reaches the maximum $H(\mathbf{y}) = \log n$.

Definition 12 (Distributedness of presentation): The distributedness of area A is the probability $\mathrm{P}(\mathbf{y})$ weighted distributedness of $\mathbf{y} \in A$:

$$\sum_{\mathbf{y} \in A} H(\mathbf{y}) \mathrm{P}(\mathbf{y})$$

As we can see, the distributedness of retina is high, since it is highly unlikely that only one receptor fires. Likewise, the motor area of vocal tract is also highly distributed, since many motors fire together to speak or sing.

Brain areas have a varied degree of distributedness. However, each neuron does not have a symbolic meaning. Solution: at a different motor context, each hidden neuron has a different meaning, because each hidden neuron has not only bottom up inputs but also top-down inputs. Top-down inputs are from multiple motors each having different meanings.

Theorem 12 (Nonuniqueness of meanings): Each hidden neuron does not represent a unique meaning.

Proof: The proof follows from that of Theorem 10. Each hidden neuron has not only bottom-up input (which is often concrete), but also top-down input (which is often abstract). Thus, under a different motor (context) input, each hidden neuron has a different meaning. ■

XIII. The symbol problem

Definition 13 (Symbol): A symbol is a token that a human programmer uses in a design document to specify a meaning to other humans typically described in a natural language.

Definition 14 (Emergence): A representation is emergent if it is directly generated from sensors (including biased sensors, such a pain and sweet sensors), motors, or hidden areas that take inputs from sensors and motors.

Definition 15 (Symbol problem): Symbols do not ground the world automatically at any time (the grounding problem) nor do they ground across times (the frame problem).

Solution to the symbol problem: a network (e.g., DN-2) that uses emergent (nonsymbolic) vectors in system representations.

This solution implies that symbol grounding (binding) proposed by Stevan Harnad 1990 [35] and many others is flawed.

Theorem 13 (Symbolic impoverishment): If a machine does not learn many longer and higher contexts beyond a whole set of taught symbols, the set of symbols is impoverished for the system to be conscious of all the taught symbols while processing sensory responses and motor responses.

Proof: The system is not conscious of all the taught symbols according to Definition 5 as the system is not aware of "longer and higher" contexts of these symbols. ■

XIV. The local minima problem

Definition 16 (Local minima): While searching for weight parameters and system hyper-parameters, starting from a different set of the parameters leads to a different local minimum in an objective function. This is called the local minima problem.

Error-backprop learning [16], [36] gives very different performances for trained networks from different initial weights. Solution: a network (e.g., DN-2) that use adaptive competitions that avoid error-backprop. See Sec. XIX about how the optimality solves the local minima problem.

Greedy methods such as error-backprop get stuck into local minima, depending on which initial seed is used for weights. Weng [19] explained why CNNs, LSTM, ELM and many other networks suffer from a severe problem of local minima. Post-Selections cannot give a single system and only report the luckiest system among many trained systems. Many unreported trained systems did worse or a lot worse. At least the average error of all trained systems should be reported, instead of the minimum error. This means that many published performance data are grossly inflated and should be re-examined.

One may say: As long as I find a system that works for a data set, do not blame me to cast dice. The following definition is meant to answer this type of arguments.

Definition 17 (Dice problem): A technique suffers from a dice problem if it requires a process of casting dice $n > 1$ times and n systems are trained and tested in Post-Selection.

A customer of such a technique faces an unacceptable uncertainty and risks, because it is typically unrealistic for the customer to test $n > 1$ systems. For example, for driverless cars, failed systems would crash the car and injure people.

It is important to note that the ML optimality is fundamentally different from a network that aims to only minimize a nonlinear objective function, such as with error-backprop methods. As further discussed in Sec. XIX, the ML optimality finds the closed-form solution for the observed distribution of a huge number of parameters in the hidden representations.

Error-backprop does not estimate such distributions at all.

XV. The abstraction problem

Scholars like Michael Jordan complained neural networks do not abstract well [11], [37]. However, the abstraction problem of neural networks has not been clearly defined. The adjective "abstract" is defined as something that is not concrete or not physical. Let us define the abstraction problem here.

Definition 18 (Abstraction problem): The abstraction problem for an intelligent system, natural or artificial, is to produce abstract properties from a physical or concrete example.

Examples of abstract concepts include location, type, scale, relation, and rule. Physical examples have a concrete combination of values in such abstract concepts (e.g., particular values of location, type, scale, relation and rule). An abstract concept (e.g., location) must be invariant with (or disassociated with) any other concepts (e.g., type, scale, relation, and rule).

For example, features in a CNN [13], [16], [36] are always location-specific (centered as a particular location of a convolution kernel) and therefore does not abstract for location. If one wants to produce a location value (e.g., along with segmentation) from the CNN, a separate process, in addition to the CNN, is required, as Cresceptron [13] did.

Solution: a convolution-free network (e.g., DN) using a hierarchy of feature representations that fluidly bridges the sensor end (which is concrete) and the motor end (which is abstract) [38]. This solution should address all "abstract" concepts, including consciousness.

Theorem 14 (Exponential problems): If symbolically approached, the recognition/detection/segmentation problem from a clutter scene is exponential in computational complexity.

Proof: For each object prototype, consider three factors.

1) Segmentation of a part: each pixel has c color-intensities, a part with e pixel-elements has $O(c^e)$ complexity. The "Big O" notation $O(f(n))$ denotes the upper found on growth rate of $f(n)$.
2) Group flexible parts into a prototype object: Each part is centered at location l, the number of combinations of p parts of a prototype is $O(p^l)$.

3) Segment the prototype from a cluttered background: Suppose a cluttered scene has m parts, $m \gg p$. Segmenting the prototype from a cluttered scene (many parts!) has a complexity $O(2^m)$ where 2 is for inside prototype or outside prototype.

The real complexity per prototype is a product of above three exponential complexities: $O(c^e p^l 2^m)$. ■

John Tsotsos [39, Fig. 4] assumed a prototype based search that amounts to $O(2^{em})$ for each prototype without probably considering flexible parts $O(p^l)$ which is also exponential.

Theorem 15 (Abstraction from exponentials): The hierarchy in Theorem 10 abstracts from features in hidden Y area using a constant resource in memory and time, without dealing with the exponentials in Definition 14.

Proof: In DN, at each time t the context in the motor area Z predicts the active feature neurons in the hidden area Y in a constant frame time, as shown in Eq. (1). This is the power of numeric interpolation that avoids any symbolic formulation. Further details of the proof are available in [21]. ■

The motor output from a scene is not unique and is always task dependent and context dependent. A different context may lead to a different motor explanation (e.g., "Rubin's vase" and "My Wife and My Mother-in-Law").

XVI. The compositionality problem

Compositionality, defined vaguely in philosophy, states that the meaning of reading or writing a sentence is not present in the words of the sentence, but metonymy is required. Proponents of compositionality typically emphasize the productivity and systematicity of our linguistic understanding [40]. Meanings lie in life experience, namely consciousness.

Solution: a network (e.g., DN-2) applies lifetime experience as sentence meanings during conscious learning [1], [2].

XVII. The smooth representations problem

Representations in a brain are globally smooth in 3D. Solution: a network (e.g., DN-2) in which the global smoothness is maintained from small to large so that newly spawn neurons are near their parents, and start with the parameters of the parent [20], [41], namely, coarse-to-fine approximation.

XVIII. The motivation problem

A brain takes reinforcers and has various emotions [42], [43]. Solution: a network (e.g., DN-2) bootstraps motivation through four classes of neural modulators: serotonin (5-HT) which represents pains, dopamine (DA) which represents sweet, acetylcholine (ACH) which represents expected uncertainty and norepinephrine which represent unexpected uncertainty.

Different neural transmitters have different effects to different neurons, e.g., resulting in (a) avoiding pains, seeking pleasures and speeding up learning of important events and (b) uncertainty- and novelty-based neuronal connections (synaptic maintenance for auto-wiring) and behaviors (e.g., curiosity). Thus lower motivations develop higher motivations, emotions and goals throughout lifetime, including the levels 1 to 5 in consciousness.

XIX. The optimality problem

The brain learns incrementally but must avoid catastrophic memory loss [44]. Solution: a network (e.g., DN-2) that is optimal in the sense of maximum likelihood [19] across lifetime. The ML-optimality is more general and more powerful than solving the local minima problem.

Let us recall the definition of a maximum likelihood estimator (ML) from a batch data. Let $\mathbf{x}$ be the observed data and $f_\theta(\mathbf{x}, \mathbf{z})$ be the probability density function that depends on a vector θ of parameters. The ML estimator for θ corresponds to the θ that maximizes the probability density. The vector θ contains all parameters in the network, including the hidden Y area. Regardless $\mathbf{z}$ is imposed or not, $\mathbf{z}$ is part of the parameters to be computed as self-generated version:

$$(\theta^*, \mathbf{y}^*, \mathbf{z}^*) = \operatorname*{argmax}_{(\theta, \mathbf{y}, \mathbf{z})} f_\theta(\mathbf{x}). \tag{3}$$

Since lifetime estimator is incremental, at each time t, the previous state $\mathbf{z}^*_{t-1}$ is self-generated or rarely supervised, and the observation is $\mathbf{x}_{t-1}$. The incremental ML-estimator for θ^*_t is computed by the incremental version of Eq. (3) where f uses context $\mathbf{c}_{t-1} = (\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_{t-1})$:

$$(\theta^*_t, \mathbf{y}^*_t, \mathbf{z}^*_t) = \operatorname*{argmax}_{(\theta_t, \mathbf{y}_t, \mathbf{z}_t)} f_{\theta_t}(\mathbf{x}_{t-1}, \mathbf{y}^*_{t-1}, \mathbf{z}^*_{t-1}). \tag{4}$$

The DN computes the above expression for each time t in a closed form without conducting any iterations. This closed form solution ML-optimally realizes f_i in Eq. (1), for $i = 0, 1, 2, ...$

Different random seeds at $t = 0$ result in different initial weights, but result in the same DN network as explained in [44]. Thus, each DN computes only a single but ML-optimal network, without a need for Post-Selections.

It is important to note that ML-optimal learning here is fundamentally different from "error minimization" in CNN, LSTM and ELM, etc. All such trained networks do not estimate the distribution of a huge number of parameters. Thus, they are all local minima, actually very bad local minima because their error-backprop does not have competition as Weng [19] argued.

XX. The auto-programming for general purposes (APFGP) problem

This problem is too extensive to be fully explained here. The reader is referred to Weng 2020 [45] for a complete exposition. A brain must learn to write a complex program for any practical purpose. Solution: a network (e.g., DN-2) learns a universal Turing machine. However, robots have not yet reached an adult age that is necessary to have sufficient mental skills. Future Conscious Learning robot would read and write books, design dramas and make business plans.

XXI. The brain-thinking problem

Over 70 years ago, Alan Turing [46] raised a now well-known question: "Can machines think?" The problem is too extensive to be fully explained here. See Wu & Weng 2021 [47] for a complete theory and some experiments about machine thinking for general purposes, using brain inspired DN-2.

A brain conducts thinking like planning and discovery. Solution: a network (e.g., DN-2) conducts brain-like thinking including planning and discovery [47] for any practical purposes.

XXII. Conclusions and Discussions

There are other important problems. Due to space limit, we discuss 20 of them. It seems that modeling biology at the brain scale is not only necessary for understanding a brain, but also critical for solving many engineering problems like those discussed here. It seems that an effective solution to any of the 20 problems is impossible without solving them holistically by conscious learning, as reported in Weng 2020 [1], [2].

References

[1] J. Weng, "3D-to-2D-to-3D conscious learning," in *Proc. IEEE 40th International Conference on Consumer Electronics*, (Las Vegas, NV, USA), pp. 1–6, Jan. 7-9 2022. http://www.cse.msu.edu/~weng/research/ConsciousLearning-ICCE-2022-rvsd-cite.pdf.

[2] J. Weng, "An algorithmic theory of conscious learning," in *2022 3rd International Conf. on Artificial Intelligence in Electronics Engineering*, (Bangkok, Thailand), pp. 1–10, Jan. 11-13 2022. http://www.cse.msu.edu/~weng/research/ConsciousLearning-AIEE22rvsd-cite.pdf.

[3] L. R. Squire, J. L. Roberts, N. C. Spitzer, M. J. Zigmond, S. K. McConnell, and F. E. Bloom, *Fundamental Neuroscience*. San Diego, California: Academic Press, 1999.

[4] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, eds., *Principles of Neural Science*. New York: McGraw-Hill, 4th ed., 2000.

[5] WSTDF, "Closing ceremony," in *Proc. the 3rd World Science and Technology Development Forum*, Beijing China: WSTDF, Nov. 7, 14:00-16:40 2021.

[6] J. Weng, "Symbolic models and emergent models: A review," *IEEE Trans. Autonomous Mental Development*, vol. 4, no. 1, pp. 29–53, 2012.

[7] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001.

[8] D. B. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[9] D. B. Lenat, G. Miller, and T. T. Yokoi, "CYC, WordNet, and EDR: Critiques and responses," *Communications of the ACM*, vol. 38, no. 11, pp. 45–48, 1995.

[10] M. Minsky, "Logical versus analogical or symbolic versus connectionist or neat versus scruffy," *AI Magazine*, vol. 12, no. 2, pp. 34–51, 1991.

[11] L. Gomes, "Machine-learning maestro Michael Jordan on the delusions of big data and other huge engineering efforts," *IEEE Spectrum*, Online article posted Oct. 20, 2014.

[12] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Trans. Systems, Man and Cybernetics*, vol. 13, no. 5, pp. 826–834, 1983.

[13] J. Weng, N. Ahuja, and T. S. Huang, "Learning recognition and segmentation using the Cresceptron," *International Journal of Computer Vision*, vol. 25, pp. 109–143, Nov. 1997.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

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

[18] G. B. Huang and C. K.Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[19] J. Weng, "On post selections using test sets (PSUTS) in AI," in *Proc. International Joint Conference on Neural Networks*, (Shengzhen, China), pp. 1–8, July 18-22 2021.

[20] X. Wu and J. Weng, "Neuron-wise inhibition zones and auditory experiments," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9581–9590, 2019.

[21] J. Weng, Z. Zheng, and X. Wu, "Developmental Network Two, its optimality, and emergent Turing machines." U.S. Patent Application Number: 16265212, Feb. 1 2019. Approval pending.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[23] J. Weng, "Brain as an emergent finite automaton: A theory and three theorems," *International Journal of Intelligence Science*, vol. 5, no. 2, pp. 112–131, 2015.

[24] G. Bi and M. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual Review of Neuroscience*, vol. 24, pp. 139–166, 2001.

[25] J. Weng and M. Luciw, "Dually optimal neuronal layers: Lobe component analysis," *IEEE Trans. Autonomous Mental Development*, vol. 1, no. 1, pp. 68–85, 2009.

[26] I. Merriam-Webster, ed., *Merriam-Webseter's Online Dictionary*. Springfield, Massachusetts: Merriam-Webster, 2022.

[27] E. H. Hess, *Imprinting: Early Experience and the Developmental Psychobiology of Attachment*. New York: Van Nostrand Reinhold Company, 1973.

[28] D. Pierce and B. J. Kuipers, "Map learning with uninterpreted sensors and effecters," *Artificial Intelligence*, vol. 92, pp. 169–227, 1997.

[29] J. Bonvillian, K. Nelson, and V. Charrow, "Language and language related skills in deaf and hearing children," *Sign Language Studies*, vol. 12, pp. 211–250, 1976.

[30] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, pp. 139–160, 1991.

[31] J. A. Starzyk and D. K. Prasad, "A computational model of machine consciousness," *International Journal of Machine Consciousness*, vol. 3, no. 2, pp. 255–281, 2011.

[32] D. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[33] M. Luciw and J. Weng, "Laterally connected lobe component analysis: Precision and topography," in *Proc. IEEE 8th Int'l Conference on Development and Learning*, (Shanghai, China), pp. 1–8, June 4-7, 2009.

[34] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge, UK: Cambridge University Press, 1990.

[35] S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, pp. 335–346, 1990.

[36] Y. LeCun, L. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[37] M. I. Jordan, "David rumelhart memorial plenary talk." Delivered at the International Joint Conference on Neural Networks, 2011.

[38] J. Weng and M. D. Luciw, "Brain-inspired concept networks: Learning concepts from cluttered scenes," *IEEE Intelligent Systems Magazine*, vol. 29, no. 6, pp. 14–22, 2014.

[39] J. K. Tsotsos, "Analyzing vision at the complexity level," *Behavioral and Brain Sciences*, vol. 13, pp. 423–469, 1990.

[40] Z. Szabo, *Problems of Compositionality*. New York: Routledge, 2000.

[41] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett, *Rethinking Innateness: A connectionist perspective on development*. Cambridge, Massachusetts: MIT Press, 1997.

[42] H. L. Dreyfus, *What Computers Still Can't Do*. Cambridge, Massachusetts: MIT Press, 1992.

[43] J. Weng, *Natural and Artificial Intelligence: Introduction to Computational Brain-Mind*. Okemos, Michigan: BMI Press, 2nd ed., 2019.

[44] J. Weng, "A developmental method that computes optimal networks without post-selections," in *Proc. IEEE International Conference on Development and Learning*, (Beijing, China), pp. 1–6, August 23-26 2021.

[45] J. Weng, "Autonomous programming for general purposes: Theory," *International Journal of Huamnoid Robotics*, vol. 17, pp. 1–36, August 2020.

[46] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, October 1950.

[47] X. Wu and J. Weng, "On machine thinking," in *Proc. International Joint Conf. Neural Networks*, (Shenzhen, China), pp. 1–8, IEEE Press, July 18-22 2021.