Abstract—The question how neural systems (of humans) can perform reasoning is still far from being solved. We posit that the process of forming Concepts is a fundamental step required for this. We argue that, first, Concepts are formed as closed representations, which are then consolidated by relating them to each other. Here, we present a model system (agent) with a small neural network that uses realistic learning rules and receives only feedback from the environment in which the agent performs virtual actions. First, the actions of the agent are reflexive. In the process of learning, statistical regularities in the input lead to the formation of neuronal pools representing relations between the entities observed by the agent from its artificial world. This information then influences the behavior of the agent via feedback connections replacing the initial reflex by an action driven by these relational representations. We hypothesize that the neuronal pools representing relational information can be considered as primordial Concepts, which may in a similar way be present in some pre-linguistic animals, too. This system provides formal grounds for further discussions on what could be understood as a Concept and shows that associative learning is enough to develop concept-like structures.

Index Terms—Artificial agent, feedback connectivity, reasoning system.

I. INTRODUCTION

HOW TO clearly distinguish processes of reasoning and deliberation from reflexes in animals is unclear and it is also unknown at what level in the animal kingdom the former begins to emerge [1]. Evolutionary origins, as well as the organization of such processes in the human brain, are largely unknown, too. Therefore, currently, we have no means to equip robots and AI systems with such processes in a way that scales up to the level of human proficiency. In this article, we are concerned with the question how deliberation could begin to emerge in a simple network from low-level neural principles. We posit that such processes must be fundamentally linked to the formation of Concepts in an agent upon which deliberation can commence. The goal of this article is to show that small networks can perform the first small steps into this direction. Hence, we want to arrive at an algorithmically implementable theory that uses realistic neuronal operations working toward Concept formation.

The nature of Concepts is a subject of a long debate because it can be approached from different perspectives: e.g., language, psychology, and neuroscience, but possibly it is studied most extensively in philosophy [2]. Concepts are frequently explained through examples given in natural language and are considered to be “mental representations” [3]–[5]. However, such approaches are rarely mathematically precise nor do they suggest a procedure for algorithmic implementation. For our approach, we get inspiration from the process of Peircean semiosis [6], which is among the most constructive approaches in this field and leads us toward an algorithmic implementation.

According to Peirce [6], semiosis has three elements which are Sign, Object, and Interpretant. Their triadic relationship is shown in Fig. 1. Sign represents a structure in the domain of perceivable signals, for example, letters, utterances, etc. Object is the entity to which the Sign refers, whereas Interpretant is the effect of a Sign on an interpreter (e.g., a person) who perceives the Sign. This triadic structure captures what we nowadays call a “Symbol” (e.g., in the context of the development of communication between artificial agents).
Semiotics is a complex multidisciplinary field and we cannot provide an in-depth discussion here. Rather than that, we would like to ask about the neuronal processes underlying semiosis.

For our purposes, one aspect in this discussion remains simple. The actual Sign will not matter. Call it cat, Katz, ξας, chat, or xyz, as long as the interpreter (and possible communication partners) know which object concept stands behind these Signs all will be good. Here, we address the hard problem to define the Interpreter. We ask the question, what is the Concept underlying the Object to which the Sign “ξας” refers? Hence, we are concerned with the right side of this triangle that creates the Concept.

The central problem addressed in this study is how to arrive at a primordial neuronal system that displays some of the processing steps needed for the formation of Concepts. Underlying this is the question, how advanced symbol-processing engines—humans—could have evolved from their ancestors. We would like to show here some simple neuronal processing steps that might belong to those that have kick-started the evolution toward advanced symbol processing and language.

We are here considering early (potential) stages in evolution. Thus, we are concerned with autopoietic systems [7], because such agents were not yet supervised from “someone else.” Thus, all learning therein should be based only on the statistical properties of the world. Such a system should be designed so that feedback comes only from the environment through the system’s own behavior and own sensing capabilities [8].

Hence, our system should carry within its activation patterns some characteristics that could be interpreted by an external observer as “going toward representing a Concept” and it should not just follow a chain of input-triggered and reflex-like actions to achieve a goal. Nonetheless, these “Concept-like entities” should arise solely from sensory inputs. As a consequence of this, the behavior should then be triggered when certain activity patterns arise from the “Concept assemblies” and feed back to lower layers that are concerned with sensory-motor processing. The central problem that immediately shies up here is how to get from plain sensory signals to Concepts and back to motor signals. The heavy use of quotation marks above indicates that this primordial system will still remain quite far away from the symbol-handling capabilities found in apes, let alone humans.

To summarize, the contribution of this study is an artificial neural system that:

1) forms Concept-like entities based on sensor (specifically, visual) inputs using associative learning;
2) interprets the developed Concept-like entities by performing (virtual) actions appropriate for the given environmental condition;
3) allows quantitative evaluation of performance through the evaluation of action appropriateness.

From a machine learning perspective, our emphasis is on the problem of signal-to-symbol transformation based on associative learning. We argue that concept formation is an essential step in symbol formation. We hypothesize that concept formation can be bootstrapped by associative learning, given input from the environment. We prove this hypothesis by showing that the system can behave appropriately based on concept-like entities, developed in an associative way. Admittedly, much of our conclusions will be speculative and a matter of debate (see also Section V), which we would like to stimulate with this article.

II. STATE OF THE ART

**Symbol Versus Signal:** Deliberation processes, as considered in this study, are most strongly related to language or classical AI, both operating on discrete symbols [9]. However, this cannot be directly used in agents acting in the real world (e.g., robots), because both, sensors and motors, rely only on continuous (analog) signals. Thus, to use symbols with robots one needs a signal-to-symbol transformation. This problem had been addressed since the early days of AI in image- and other signal-processing applications in order to achieve a human-like analysis of a visual scene or situations derived from other (nonvisual) signal information [10]–[12]. Methods for covering the gap between signals and symbols many times are analyzed from the perspective of symbol grounding [13]; that is the process of meaning acquisition of a symbol, where “meaning” itself is a complex concept [14]. In robotics, symbol grounding is important with respect to several aspects. First and most straightforward, there is the aspect of communication with (or amongst) artificial agents where grounding of communicative symbols is needed for the agents to understand each other [15]. The second aspect is less straightforward and associated to the usage of layered architectures in cognitive robotic systems, where at the lowest layer sensorimotor signals are considered and at the highest layer, symbolic planning is performed [16]. Thus, symbols first need to be derived from sensor signals to supply the planning procedures and then symbols coming from the planner need to be translated to (voltage) signals for robotic motor actions. All in all, this comprises a pathway: signal-symbol-signal. Intriguingly, the second aspect is associated to “speaking to yourself,” discussed in the philosophical literature [17]. One can imagine a human mental plan to be a sequence of steps (tokens) kept in one’s mind and communicated to yourself and (often) leading to (muscle) actions. Thus, this links aspect two (layered architectures) back to aspect one (communication).

Symbol grounding in artificial agents is frequently addressed using multimodal categorization in sensorimotor space and statistical inference on top of that to associate the developed categories with words, i.e., symbols (e.g., told by a human) [18], [19]. Such methods are applied in developmental robotics where robots can perform externalization of their perception and communication within the learned domain [19], [20]. However, until now only limited complexity is reached [21], [22]. For robots to perform more complicated tasks, instead of grounding processes through self-experience, symbol anchoring [23] is supplied by the system’s designer in a supervised domain-based manner. Anchoring concentrates on creating and maintaining the relation between symbolic and sensory information, while grounding concentrates on the more advanced aspects of creating the meaning of
symbols [24] from first (sensori) principles. Thus, anchoring is more suited for robotic applications within well-defined domains where the designer has “all” knowledge and the robot never stumbles upon anything unexpected by the designer (which it would, thus, not “understand”). Hence, in such architectures, appropriate procedures are defined for world state estimation from sensing and planning where preprogrammed actions are then triggered [25], [26]. Clearly, these approaches have only limited generalization abilities, mostly based on a reshuffling of the developed structures [27], [28] where also repurposing of the structures using high-level reasoning can be achieved to some degree [29], [30].

Connectionism and Symbols: An alternative way to arrive at situation-matched actions, as compared to classical AI, is by employing end-to-end learning (sensor to action), which is currently widely investigated [31], [32]. However, such approaches also do not reliably generalize to new situations and are hard to interpret, because sensor-to-action networks are difficult to comprehend. This none withstanding, already Harnad [13] had advocated that connectionism—hence the use of networks—can serve for categorical representation using bottom-up approaches. Furthermore, we are a living proof of networks that use symbols: humans do reason symbolically and can manipulate symbols in an abstract way using their brain networks. But what makes us different from animals that cannot do this? Penn et al. [33] have suggested that it is the handling of relations between and on top of other relations that distinguishes the human from an animal’s mind. This would correspond to a so-called physical symbol system [34] which considers abstract symbol manipulations of the type used in computers (and classical AI). However, using a neuronal implementation to target full-fledged abstract symbolic reasoning is still too ambitious for our currently existing artificial networks and it remains still too complex to attempt symbol grounding this way starting from sensory processing. Due to that, physical symbol systems are not much considered in recent studies of symbol emergence.

Concepts as Units of Thought: Even if one does not target to explain the animal-to-human cognition leap [33], one still can take inspiration from human thought processes and researchers debate how human thought is organized (e.g., does language come first or, alternatively, other aspects of reasoning, e.g., nonlanguage based concepts?) [5], [35]. The term “Concept” is here many times used to define structural units of thought. This term is mostly applied in philosophy, psychology, and linguistics, whereas introducing this term (and associated properties) into the realm of artificial systems is a relatively recent development. This, however, might be needed if one wants to organize an artificial system such that the processes therein resemble human thought.

In philosophy, Concepts are approached from several different perspectives, where the main trends are: mental images, abilities, or abstract objects [2]. Philosophers talk about mental representations arising through experience since centuries (see [36]). In the mental image approach, Concepts are considered as building blocks of thought, where more complex representations are composed of simpler ones and where at the bottom reside primitive representations with perceptual character [37]. This, in principle, reflects the grounding processes addressed in computer science. Critique to the mental image approach arises due to difficulties in connecting abstract Concepts to perception. Alternative approaches span the domains of language and logic [3], [38], [39], while assigning less importance to perception. We cannot provide an exhaustive discussion of all these aspects here. However, it is important to note that philosophical approaches can only address—in a consequent way—explicit facts and knowledge that is represented with language-based argument, whereas for analyzing implicit representations in neuronal systems one needs different methods, too (mathematical or computational modeling), not offered by philosophy.

Partially leveraging from philosophy and linguistics is semiotics [40], with its emphasis on the relation between meaning and signs. Here, the origins of meaning as well as questions on syntax and semantics, and the usage of both are addressed. Thus, semiotics is instructive for the formal modeling of the acquisition of meaning. We have already explained above the triadic structure of Peircean semiosis, involving Object, Sign, and Interpretant [41], [42], which is also part of the basis of our study. We follow these views, because the relation between the Object and its interpretation by an agent, the right side of the triangle, is of primary importance here.

Psychologists experimentally test hypotheses about the categorical and conceptual organization of the human mind. Rosch [43] found out that for humans not all representatives attribute equally to a particular category (e.g., apples are considered to represent the fruit-category better than plums or oranges). This is different from the classical view of philosophers stating that all representatives of the same Concept are equal. Psychologists, thus, adhere to the Prototype Theory that states that Concepts are representations whose structure is based on statistical analyses of properties of their members [37]. Another frequently used conjecture is that Concepts are mental representations embedded in structures called theories. Thus, Concepts need to be analyzed in relation to other Concepts. Psychology analyzes how such theories are built during ontogeny [44]. By now evidence exists that abstract Concepts do have associations to sensory processing [45], [46]. Mirror neurons are potential candidates of conceptual representation at a fine scale [47], [48].

Biolinguistic theory investigates the course of evolution (and evolutionary leaps in particular) in attaining attributes allowing complex conceptual organization in the brain as well as capabilities to externalize that, using language. However, the current line of thought here is that animals already have rich combinatorial thought, even though externalization might first be rather action than language based [49], [50], which matches our approach taken in this study.

In robotics, the usage of the term Concept is rather new and not yet standardized [51]–[53]. Taniguchi et al. [53] introduced some more clarity for the usage of the terms Category, Concept, etc. and suggest that Concept is an internal representation “that goes beyond the feature-based representation of Category” and that the semantic essence of the Category needs be extracted to arrive at a Concept.

Cognitive Entities in Artificial Neural Networks: Some works addressing higher cognition using neural networks also exist, but these systems have a different aim and have almost
no intersection for comparison with our study. Hummel and Holyk [54] developed a network-based cognitive architecture for acquiring schemas up to an ability to process-relational representations, this way addressing human-like cognition [33]. However, their architecture does not reach down to the sensory level and thus, does not perform grounding of the representations. Sandamirskaya et al. [55] developed neural representations in sensorimotor space based on dynamic field theory, which shows emergent behavior that has qualities of higher order cognition. However, these systems are developed for demonstrating simple behaviors in a grounded way, which are meticulously pretrained and, in addition, they need representations spanning the entire input and output range (e.g., all possible colors and space coordinates as well as all gaze directions), which makes representations relative bulky. Tani [56] associates behavior and language learning in a continuous neurodynamical system trained in a supervised manner using predefined error backpropagation schemes. Similar to that, Olier et al. [52] used variational recurrent networks for associating different modalities to obtain the situated behaviors of an agent. Xing et al. [57] developed a network mimicking different brain areas where the integration of different sensory modalities for Concept acquisition is the main target. Possibly the study closest to ours is by Ozturkcu et al. [58] investigating if neurons with meaningful (for a human) representations could emerge in small neural networks in the course of sensory-motor reinforcement learning. However, the semantics of the developed representations for the agent itself had not been addressed. Taken together, neuronal implementations grounded in sensory inputs and clearly targeted at helping to pin down the concept of Concepts do not seem to exist in the current literature.

A. Definitions: Categories and Concepts

From the above, it is evident that there is still no neuronally implementable way existing to arrive at Concepts. To approach this, we will first discuss how we would see the difference between Category and Concept and present a view that allows for partial implementation.

Following [53], we posit that Categories are unreflected clusterings within a subsystemic feature space, whereas Concepts are the narrative, the rule, and the essence of the thereby captured entities. As humans, we can speak out about (i.e., externalize) concepts using language, e.g., we say: this Concept is A and not B, it contains C but not D and it relates to E, etc. These narratives or “rule sets” for a Concept are in essence relational and, thus, clearly go beyond mere “feature constellations,” which you could obtain by clustering. The aspect of “relations” is, thus, in our opinion essential for Concept and we will build our model of Concept formation based on the following definitions (see Fig. 2, examples are given below to make this clearer).

1) Categories can be understood as clusters. They are defined only through (often pairwise) local similarity operations between their members and are often represented by a central exemplar.

2) Concepts are formed by relations and Categories are used as the basis to form them. In detail, they are as follows.

a) Nascent Concept: The first relation that is needed to bootstrap Concept formation is closure (i.e., the inclusion versus exclusion operators) obtained by drawing the boundary around clusters.

b) Consolidated Concept: Concepts are consolidated by relating them to other Concepts.

c) Externalizable (Final) Concept: Finally Concepts must entail some kind of agent-owned explicitness. The consequences of acting according to a Concept need to be observable (or self-observable). Arguably, self-observation is one central step for the forming of a mental image that represents then “this” Concept.

B. Examples

Categories as Clusters: Let us consider a set of vectors in RGB space. One could now use an algorithm that sorts this set according to statistical correlations between the vectors into a map-like representation. For example, this would work with the self-organizing map (SOM) algorithm [59]. Similar colors would this way be placed near to each other with continuous transitions from one to the other. The algorithm would this way define clusters that can be seen as color categories. But it would not be able to tell that “this is red and this not.”

Nascent Concepts via Closure: Closure is fundamentally needed to be able to formulate a relation that says this is red and this is not. Hence, you need to (mentally) draw a boundary enclosing all points according to your idea of which color is (still) red and which is not. This is the first step toward Concept and requires the nonlocal process of “drawing this line,” i.e., defining a decision boundary.

Consolidated Concepts From Using More-Complex Relations: Closure can only create a narrative that says
Fig. 3. Architecture of the system and signal flow. Convolutional layers 4–6 of a pretrained neural network are used as feature reservoir (gray box) for training network on the right (dashed box) using associative learning. The small neuron icons in Big 3 and in the convolutional layers indicate that these neurons use two dendrites each for learning. (a) Feedforward: After training, the network in the dark green box triggers the action “run” of a virtual agent in case the object in the image is big. (b) Feedback: A small yellow item on the left (which is really an occluded big one) triggers “running” via feedback. Fat arrows: black=feed forward, violet=feed forward after AND operation, and red=feedback. Note, the action in our system is virtual, represented merely as a signal for run, which is used for the evaluation of the system. Inset in (a) shows the examples of input images.

“belongs to” versus “does not belong to.” Human concepts are usually far deeper than that and rely on many more relations. For example, more complex relational operations would in this case be: [a (traffic light) is (composed of) (green), (yellow) and red] or [red is (on top of) the (traffic light)]. Where [ ] refers to the relation used to explain the Concept “red” and ( ) refers to other Concepts underlying these respective relations. Clearly, entities in ( ) are Concepts in their own right and would have to have their own explanations without which [ ] would be incomprehensible. Here, we must limit ourselves to addressing simple relations at the level of propositional logic as our system cannot attempt explaining more complex human concepts.

Externalizable (Final) Concept by Agent-Owned Explicitness: Representations within a brain (or artificial neural network) that capture all aspects above but just exist without triggering any behavior are—at best—“incomplete” Concepts. Concepts need to entail behavior in a closed-loop context. Note that human mental simulation is here subsumed also under “behavior.” Also note that externalization comes for humans in several stages. For example, children at the age of about three can correctly respond to the request: “Please give me the red pencil!” but they cannot yet answer the question “Which color does that pencil have?” (in spite of using, in other situations, the word red!). They have, we would say, only a partially externalizable Concept of red. While they must be able to form a mental picture of “red pencil” to be able to react to the request, the final narrative stage (discourse or self-discourse) has not yet been reached at that age.

This allows us to consider how an artificial neural system could arrive at Concepts. We would need neurons that perform 1) closure and represent 2) relations between entities in their synaptic weights. Ideally, the system should then 3) externalize the Concept and create behavior and finally a narrative. The system presented below will show properties 1 and 2 albeit only for some simple cases. Property 3 will emerge, too, as the system will produce responses (after Concept formation) for “imagined situations” as if it were forming some kind of mental image, like our 3-year old child from above. But narratives are not yet being made by this system.

Clearly, our system will not even get close to the complexity of a 3-year old. However, we believe that the aspects shown below, all of which are biologically realistic and generalizable (independent of input modalities), may help to plant the cognitive processes for Concept formation better in some algorithmically sound and transferable soil.

III. METHODS

A. Basic Model Structure and Goal—Overview

The goal of our network is to emulate an agent that should learn to react in an appropriate way to certain Concepts, which arise from visual signals.

To achieve this, the systems consist of two fundamental components. There is a conventional convolutional neural network (Fig. 3(a) and (b) left, gray) pretrained once in a supervised manner on a surrogate task and used as feature reservoir. This could be considered related to the early visual areas (e.g., V1) in the visual cortex [60].

In addition to this, we implemented a second network, called Acting Agent Net (AAN, Fig. 3(a) and (b) dashed). This network is indiscriminately fed features from convolutional layers 4–6 of the CNN and is trained using associative learning. First, we use reflex avoidance learning, which allows the selection of environmentally relevant input signals (and,
thus, delivers reinforcement from the environment. In the following stages, we use variants of classical Hebbian learning to process associations between those environmentally relevant signals.

The primary operation of the network is to trigger a reflex [follow black arrows in Fig. 3(a)] by which the agent responds to process associations between those environmentally relevant signals. Following stages, we use variants of classical Hebbian learning to thus, deliver reinforcement from the environment. In the following stages, we use variants of classical Hebbian learning to process associations between those environmentally relevant signals.

The primary operation of the network is to trigger a reflex [follow black arrows in Fig. 3(a)] by which the agent responds with the virtual action run in a feedforward manner, if an image with a big shape (in human terms: a dangerous entity) is processed by its early sensorial layers. The central task of the network, however, is to replace this primary reflex by learning a— in the inputs existing— (hidden) conjecture, namely, that yellow entities on the left are usually big and, thus, to learn [Yellow & Left → Big] and react accordingly.

The fat arrows in Fig. 3(b) shows how to augment the primary operation in the AAN by feedback to react to this conjecture. Briefly (for details see “specific architecture and mode of operation,” below): first an intrinsic closure process on the different visual features leads to nascent Concepts (green box). The agent then learns that certain relations between these nascent Concepts exist (light brown box), allowing it to form consolidated Concepts, which are indicative of a big entity. If that happens, feedback (dark yellow box) to the sensorial levels is provided “as if” a big entity had been visible. The running action is then triggered by this “imagined” big entity, indicative of a final, externalized (albeit prelinguistic) Concept.

### B. Data Set

For this study we use simple computer-generated RGB images with 100 × 100 pixels, depicting uniformly colored shapes with three preset features: 1) geometric shape (circle, square, and triangle); 2) color (yellow, magenta, and cyan); and 3) size (big, medium, and small). This creates 27 possible combinations. Some examples are shown in the inset in Fig. 3(a). Colors and sizes are variable within predefined non-intersecting intervals. These shapes can be placed anywhere on the canvas.

First, we train the CNN using a set of 24,300 images in total (900 × 27). This is done in a standard supervised manner only once and the CNN remains unchanged for all experiments. The output of the CNN is then able to recognize all inputs existing— (hidden) conjecture, namely, that yellow entities on the left are usually big and, thus, to learn [Yellow & Left → Big] and react accordingly.

The fat arrows in Fig. 3(b) shows how to augment the primary operation in the AAN by feedback to react to this conjecture. Briefly (for details see “specific architecture and mode of operation,” below): first an intrinsic closure process on the different visual features leads to nascent Concepts (green box). The agent then learns that certain relations between these nascent Concepts exist (light brown box), allowing it to form consolidated Concepts, which are indicative of a big entity. If that happens, feedback (dark yellow box) to the sensorial levels is provided “as if” a big entity had been visible. The running action is then triggered by this “imagined” big entity, indicative of a final, externalized (albeit prelinguistic) Concept.

### C. Specific Architecture and Mode of Operation

The system is activated by presenting images one after the other at the input. Hence, we operate in discrete time chunks. The activation from the image spreads through CNN layers and from there excites all blocks of neurons in the AAN one after the other as indicated by the thin arrows in Fig. 3, and may elicit some kind of (virtual) behavior. This mode of operation is found in a similar way in many mammals, where an input may trigger a certain motor program, which continues for a while, until the accumulation of information from the next input(s) triggers a change in behavior [61].

As mentioned above, the AAN network (dashed box) uses and evaluates intrinsic information contained in the convolutional layers 4–6 of the CNN. Layer 0 defines the basic behavior of the agent. Importantly, as we do not simulate any real agent all behaviors are only virtual. For example, if pool Big 0 in layer 0 (Fig. 3(a), top)
is activated above the threshold (see details in the Appendix), a virtual behavior called run will be elicited. For all zero layer pools, we use the input correlation learning rule (ICO) [62], which is a form of associative learning based on stimulus substitution in classical conditioning [63]. For that, the agent has to have a prewired reflex. If some sensory input in a persistent way arrives before the reflex-triggering signal, the agent learns to act on detecting that earlier stimulus, thus, learns to avoid the reflex. Thus, the reflex-triggering signal acts as an error term that drives the learning. This is associated to the function of the cerebellum (reflex avoidance learning, see review by Dean and Porrill, [64]). Hence, for all zero layer pools, we have pairs of “reflex-triggering” and “earlier” input signals. For run the reflex-triggering signal is a dangerously approaching shape, hence: before learning “looming” triggers run as a reflex. However, before it approaches, an earlier signal exists, which is that it is big, hence: after learning “big” triggers run proactively. As mentioned above, all “movements” in our model are only virtual, hence the big object’s approach (looming), too. Yellow and Left are set up in the same way but we do not analyze their stimulus input pairs explicitly.

Activity established in layer 0 travels to layers 1–3 and further. Different from layer 0, all other layers use variants of classical Hebbian learning, thus do not use error terms (see the Appendix for details). While Layer 0 extracts categories, layers 1–3 iteratively sharpen (discretize) the representations attained in layer 0 for Big, Yellow, and Left.

Above we had stated that the hidden conjecture of this world is that most items, which are yellow and “live” on the left side, usually are also big. Hence, it would be useful for the agent to learn the conjecture [Yellow & Left → Big] and react with running. Consider a scenario in the real world: it will be very useful for you to realize that this yellow (color feature), partially occluded agent at the water hole (location feature) very likely is a tiger from which you should run away, not waiting to see the whole big (size feature) beast. Thus, two more layers exist for this AND operation (light brown box, violet arrows). Yellow & Left 1 learns the AND, and Yellow & Left 2 is another layer for sharpening the results that came from Yellow & Left 1.

The association [Yellow & Left → Big] is formed in the neural population Big 3. This group of units receives multimodal input from Big 2 as well as from Yellow & Left 2. To allow learning of such multimodal inputs, we use a simulated dendritic structure with two branches each, where synaptic plasticity at each branch operates independently of the other branch. This type of local branch-specific plasticity is known from many neurons [65] and represents a powerful mechanism for nondestructive signal structuring in the brain. In our study, dendritic structures provide the maximum of the two branches to the output (thus, are active in the case at least one branch is active).

Finally, the signal from Big 3 travels back to the convolutional layers 4–6 in the CNN (dark yellow box). Hence, these layers also receive two different types of inputs, on the one hand from the lower convolutional layers and, on the other hand, from Big 3. The same dendritic structure as described above is used here, too. These feedback connections [red arrows, Fig. 3(b)] can now activate the behavior run, in case this behavior has not yet been activated in a feedforward manner. Hence, this will happen (after learning) as soon as the agent sees a yellow item on the left.

We use \( N = 300 \) neurons for each block, except for the CNN-activation block, where 4416 neurons are present and we usually use 50 connections from the previous block to the next block, which are chosen randomly in a uniform manner with very small starting weights. Only the connections to Yellow & Left 1 are 25 each, leading to 50 connections in total again. Feedback is organized so that each neuron from pool Big 3 provides connections to the CNN-activation neurons in the convolutional layers 4–6.

All neurons have sigmoidal-activation functions except in the CNN layers where ReLU activations are employed. Detailed equations for the network activation and learning rules are provided in the Appendix.

IV. RESULTS

To illustrate how the system works, first, we show in Fig. 4 one example of the activation patterns in the network obtained after learning. Activation patterns for each pool of 300 neurons are provided as averages, plotted as histograms over all images for the baseline test set (black, this is a training-set-like set with 2700 images) and the nBYL test set (red, contains 600 non-Big Yellows on the Left). All the x-axes are scaled between 0 (no neuronal activation) and 1 (maximal activation). The desired outcome, as seen in the histograms, is that the presence of a feature, for example, Big (B), in the corresponding pools (Big 1, 2, or 3), should lead to high activations, the absence (nB) to low activations.

In the initial layers 0, histograms are relatively wide with a narrow separation between two categories (e.g., Big, B versus non-Big, nB). In layers 1–3 increasingly stronger discretization is present. The Hebbian learning performed in these layers leads to a continuous sharpening of the distributions without any additional mechanism. This is instrumental for being able to perform the AND operation successfully (see Yellow & Left 1). The result of the first AND operation is then one more time sharpened (Yellow & Left 2) and transferred to Branch 2 of the dendrite. Here, learning commences creating locally the distributions as displayed on that branch.

The other branch receives the output from Big 2 and also performs learning. Results of both dendritic learning processes are then combined at the somata of the Big 3 neurons by a maximum operator. Note that a maximum operator corresponds for well-discretized signals to performing a logical OR operation common for many neural systems when operating at a low firing threshold. Thus, the output of Big 3 is close to one in case the shape is big, OR if the shape is Yellow & Left as shown in the histograms in the pink box above. This signal is fed back into the feature reservoir (convolutional layers) and modifies those through a dendritic structure similar to the one described above. Here, one branch receives the original activation from the CNN, while the second branch receives the feedback.

It is known from all deep learning networks that the activation of individual nodes is hard to interpret. This is here true, too, and feed forward as well as feedback activation in
the CNN layers is fairly nondescriptive and quite dispersed (not shown). Of essence, however, is that these activations will converge “in the right” way to layers 0 and here specifically to Big 0, which triggers the running action.

Hence, the CNN provides two types of inputs to layer Big 0. If a big item appears in an image the feedforward pathway [Fig 3(a)] will trigger the action run and we can assume that this leads to the immediate disappearance of the image from the sensorium (as the agent has turned around). Re-excitation of Big 0 via the feedback loop will have no consequence because the running action will have already commenced in this case. Alternatively, if a non-Big, yellow item appears on the left side of the image, Big 0 receives the feedback signal [Fig. 3(b)] which makes the agent run. These two cases are shown in Fig. 4 panels A (blue—feedforward activation) and B (yellow—feedback activation).

Most importantly, in the feedback case, the excitation for the set $nBYL$ has changed from close to zero (panel (a), red histogram), to close to one (panel (b), red histogram). Thus, feedback induces the effect, that even non-Big Yellows on the left strongly excite the pool Big 0, which is responsible for the action run in the model system. Statistics quantifying the discretization effects as well as about the logical conjectures, seen in Fig. 4 are presented in the supplementary material and prove that the case we are showing was not cherry picked. In addition, we show in the supplementary material for a control case where a Magenta input is used instead of Yellow, that no strong signal is produced in Branch 2 Big 3. This is because the learning rule (Balanced Hebb, see the Appendix for details) used for Branch 2 Big 3 does not increase weights in case a nonexisting conjecture is probed, which here is [Magenta & Left $\rightarrow$ Big].

Next, we analyze the errors of the system obtained in the feedforward mode [Fig 3(a)] as compared to the ones obtained after feedback [Fig. 3(b)]. We show this for the two data sets baseline and the $nBYL$ for layer zero. Our aim is to induce, using the feedback, the action run for the test set $nBYL$, without increasing the error rate for the action run on the test set baseline, where the error rate is close to zero without feedback.

Averages of error rates and standard deviations for ten trials for the two cases (feedforward and feedback, see Fig. 3) are provided in Fig. 5. For the most relevant test set $nBYL$, it can be seen that the error for Big (= run) drops substantially when feedback exists [Fig. 5(b), from about 88% to 14%].
There is a very small increase in errors for the test set baseline [Fig. 5(a)]. This is due to the fact that in the forward mode up to layer 3 still, small errors do exist in the discrimination between feature versus nonfeature. For example, look at the histogram nY in layer Yellow 3 in Fig. 4, where you can see a small peak of neurons with high activation. These small errors are fed back and lead to minor deterioration of the signal in the convolutional layers and finally to the here observed small increase in error in layers 0 after feedback [Fig. 5(a)].

Summarizing, after feedback, we have 2.23% error for action run for the baseline data set and 14.27% error for the nBYL data set. The increase in the error for correct behavior in the baseline data set is only a couple of percents, while the decrease in the error on the nBYL data set is massive, reduced by 73.88%. Thus, all in all, feedback brings new advantageous properties to the system without much disturbing the performance of the initial feedforward system.

It is, however, nontrivial to characterize the feedback as such. Feedback in our system modifies the activation of the convolutional layers, neuron by neuron, but we had argued above, that it is hard if not impossible to interpret individual activations in the convolutional layers.

However, we can ask, how “similar” is the feedback activation to the original activation and one way to show this is by calculating the correlation between feedback and feedforward activations in the CNN. We use the following procedure using the baseline test set: align the feedforward activations of the neurons in convolutional layers 4–6 into a 1-D vector \( x^f(k) = (x^f_1(k), x^f_2(k), \ldots, x^f_{4416}(k))^T \) for images \( k = 1, 2, \ldots, 2700 \) and in the same order align the feedback coming to the corresponding neurons in the convolutional layers \( x^b(k) = (x^b_1(k), x^b_2(k), \ldots, x^b_{4416}(k))^T \), thus obtaining the second vector. Then, we calculate Pearson’s correlation coefficient. This can either be done directly between the two vectors \( x^f(k) \) and \( x^b(k) \) or after averaging across a set of same-feature images (e.g., all big images): \( \bar{x}^f \) and \( \bar{x}^b \). Results are provided in Table II.

At first, we note that image-by-image correlations are far away from one, indicating that different images of the same category would (as expected) activate the convolutional layers in quite different ways. This also shows that there is not one specific image or any small set of images the feedback would “hook onto” and predominantly represent. Instead, the average of activation in image sets tells the story. Here, we obtain a very high correlation coefficient of 0.96 for Big and, thus, can state that the average feedback signal is similar to the average feedforward activation of convolutional layers obtained by big shapes. For comparison, we show in Table II correlations for medium and small shapes, too, where those correlations are much smaller.

Different from this, it is, however, important to better understand to what degree the system can extract the “hidden rule” that yellow items at left are (almost) always big. So far in the training set, we always had 15% of yellow images on the left that were violating this rule: a total of 288 yellow images exist on the left, of which 246 are big and 42 not. This produced an error of about 14% in run in nBYL test set [Fig. 5(b)]. In Fig. 6(a), we show the average percentage of running in nBYL test set when adding more and more images that violate the rule to the training set, calling them outliers. Up to a fraction of 33% outliers running behavior improves, above 33% system produces less and less running. For an explanation of the imperfect behavior of the system at a low fraction of outliers, we show in panel (b), how well layer 0 can recognize the different features, where Yellow performs less good than the others, especially at a low percentage of outliers. Hence, the orange curve is the limiting factor for the performance in (a) on the left side (see a dashed copy of this curve). The drop on the right side in (a) is explained by the fact that toward the right the rule “Yellow on the Left is most of the time Big” becomes weaker and stops existing at a fraction of 50% of “outliers,” where the outliers cannot be called so anymore. Hence, this drop in running behavior is actually desired for the right side. The green dashed curve represents a rough estimate up to which fraction the system “believes in the rule.” The inset histogram shows what happens in the transition range, where also the standard deviations in panel (a) are big. For 42% outliers we receive a bimodal distribution where the system sometimes still follows the rule with a high running percentage, but sometimes not.

Fig. 7 shows the results of a parameter analysis to demonstrate the robustness of our system. The system, we are presenting, depends on the learning rates in the different pools and on the parameters involved in the annealing of the learning (see the Appendix for detailed equations).
In Fig. 7, we show errors for baseline (gray column) and nYBL (black column) test sets. Overall, the aim of any parameter combination would be to keep performance as good as possible on the baseline test set, while getting also the best performance on the nYBL set.

Column pair 1 shows the performance obtained with the parameter set which has been used for all experiments in this article. The following comparisons show that this is indeed a good choice.

In column pairs 2–7 in this figure, we show results when we vary the learning rates for different neuron pools: layers 1–3, responsible for sharpening (column pairs 2 and 3), “AND” pool (column pairs 4 and 5) and the feedback (column pairs 6 and 7). We increase (marked as \( \times 2 \) in the figure) and decrease (marked as \( \text{div} 2 \)) the learning rates twice and evaluate the system on the two test sets after learning. The results show that errors on the two data sets vary to some degree when changing the learning rate, but the general operation of the system is not destroyed. The only larger increase in error happens when decreasing the learning rate for the sharpening subsystem (column pair 3). With increasing the learning rate for sharpening (column pair 2), the error rate for the test set nYBL drops, but then the error rate for the baseline increases, which is due to an “over training” of the feedback.

To the right of this in column pairs 8–13, we show results when we use our standard learning rate but vary the parameters that regulate the annealing (“stopping”) of the learning. Annealing starts as soon as the output of a neuron is above threshold \( \Theta_1 \). Furthermore, we also vary parameter \( c \) which regulates the speed of annealing, where a smaller \( c \) brings faster annealing. In the standard network (column pair 1), we use softer annealing \( \Theta_1 = 0.99 \) and \( c = 0.9 \) for the first sharpening layer (layer 1) and for the feedback, while elsewhere in the system abrupt annealing is used (\( \Theta_1 = 0.995, c = 0.1 \)).

The final result, however, is not very sensitive to the change in threshold \( \Theta_1 \) (column pairs 8–10). However, the learning process is destroyed in case the annealing is too abrupt with \( c = 0.1 \) everywhere (column pair 11). For column pairs 12 and 13 we used a large value of \( c \) for all pools, which makes annealing slow, but this leads to large errors in the baseline test set, indicative of over training.

However, overall, the system has a relatively wide parameter range where the performance reaches the goal of small errors on the nYBL data set, without destroying the performance on the baseline test set.

Finally, Table III provides the results of an ablation study. We connect the AND pool as well the feedback to layers 0, 1, 2, or 3, where the latter is the full system and measure errors in the two test sets. Results show that errors get smaller with adding layers 1–3.
V. DISCUSSION

A. Fundamentals of the System

We have presented the so-called AAN system that can replace an initial visually elicited reflex reaction by a feedback-triggered action which uses concept-like entities. These concepts like entities emerge through associative learning, where the first layer is receiving environmental feedback in an associative way and all other layers learn unsupervised. Visual inputs (presented as RGB images) are transformed into low-level visual features using a pretrained convolutional neural network (named CNN above), which we use only as a feature reservoir (in analogy to the early visual cortical processing areas in vertebrates). Thus, the actual structure and learning procedure of the CNN are irrelevant for this study.

Unsupervised learning first makes sure to generate activity that gets sharper from layer to layer and is high in response to this and low in response to “not this” (related to closure, layers 3). This activity is further processed by performing relational operations ([Yellow & Left] and [IF Yellow & Left \( \rightarrow \) Big]), finally triggering behavior via feedback. This feedback acts at the CNN and stimulates its neurons in a way as if a big entity had been visible. Thus, in some sense, the system “imagines” the Big entity. This together with the resulting virtual behavioral response represents a simple form of externalization.

We used different learning rules to achieve this, all of which are common in real neural systems, but details of this are not central to our study. Hence, learning rules are only presented in the Appendix. ICO-based learning [62] is used for the environmentally controlled layers of the AAN system. In other layers, Hebbian learning in some variants is used paired with an annealing procedure that limits weight growth. This mechanism is closely related to synaptic scaling [66], [67]. These two different types of associative learning are used throughout the AAN system because this allows training the system using smaller data sets and is faster as compared to deep reinforcement learning and the here-obtained network structures are better interpretable than those obtained using deep learning. Importantly, we use a dendrite-like structure to allow for calculations combining different input signals. Also, these types of structures are common in, e.g., pyramidal cells, which can perform complex and local dendritic calculations [68].

B. Evolutionary Perspective

It is arguable whether humans are the only living beings using concepts and the associated symbolic processing of reasoning (and communication). A possible indicator in prelinguistic animals—e.g., apes—for employing symbolic processing could be whether or not they can develop and execute mental plans consisting of several action steps. While planning for the future is a much agreed-upon fact in several species [69], the knowledge about which mental processes drive planning is sparse. There is still a discussion to what degree seemingly deliberate decisions might indeed be just reflex driven. The system we have developed can help in formalizing the discussion of what one would call reflex versus deliberation driven. One could ask: if one observes a type of feedback that is based on relations between discrete entities, like in our system, is this a more advanced reflex, or is this an instance of primitive reasoning? Clearly here we are still at the level of “implicit deliberation” as classified by [70] as such a system is not able to “speak about.” Furthermore, in our system, we do not yet achieve representations handling relations of other relations which is claimed by [33] to be the main characteristic of human reasoning. However, we would argue that one could use systems as the one here as a stepping stone toward building more advanced systems and use those to analyze more complicated questions about human concept formation, too.

C. Artificial System Perspective

Our system is developed in a grounded way, which is important when developing artificial agents that can act in a not predefined environment and adapt to it [71]. There are of course many grounded artificial developmental systems published [20], [72], but all are domain bound. Different from this, our system is essentially open. All mechanisms will work regardless of the type or modality of the inputs. We show how the system learns, using for concept development only the statistical structure existing in our artificial world. Hence, if there were more action-relevant intrinsic conjectures, a system with more neuronal pools could extract these without changing the general architecture.

The architecture of our system is blockwise prewired. Every block essentially represents a neuronal cell assembly with certain (learned) properties. Prewiring between blocks omits the step of also having to learn the correct forward stream for the different modalities. Hence, prewiring shortcuts this and allows extracting Concepts rather quickly. The system will, thus, in the same way, work efficiently in case of more complex conjectures, like (Yellow AND Left) AND (Triangle OR Square) if they exist in the data. Note, however, that we had shown in an older study that cell assemblies and their connectivity can also be developed by synaptic plasticity, which, however, requires prolonged self-organizing procedures [73]. Hence, doing this should work, but will be time consuming and would—in conjunction with the current study—not add to its core messages.
The development study [74] is designed to extract object categories in an exploratory way and to build planning domains based on that. The first part, extraction of object categories, is comparable to our study, but it does not put emphasis on relation formation between categories and does not analyze the distinction between categories and concepts. Furthermore, many aspects in their study are predefined and they use mostly nonbiological mechanisms (of machine learning and data storage). In general, the usage of complex predefined structures is prevalent in developmental robotics [75]–[77], whereas our system has been designed from a bottom-up sensori perspective and by employing a stronger view onto biological realism with fewer predefined aspects.

D. Limitations and Conclusion

Our system uses only excitatory connections, where more complicated tasks may also require inhibitory connectivity. Furthermore, as mentioned above, our system is blockwise prewired where we had stated that developing a system with comparable functionality from randomly connected neurons would in principle be possible. By way of analogy: human development until arrival at more and more Concepts takes years, too, and a similar effect of prolonged learning times would be obtained if avoiding the shortcut of prewiring the system to some degree.

In addition, we are here only using classical approaches to neuronal systems, stopping short of the brain’s dynamic complexity, but we show that even with this simple system we can develop models that learn primordial concepts and we think that this can serve as a valuable starting point for more complex neuronal approaches.

Our model world is (purposefully) quite simple. However, with the now-existing deep networks, it is possible to preextract decisive feature representations (beyond those of our CNN layers) when confronted with more realistic input spaces coming from the real world. Doing this might allow investigating more complicated cognitive properties (e.g., considering relations of relations which is deemed to be important for human thought, as discussed above).

Finally, in our small neural network, we do not consider making any intrinsic knowledge explicit via language. If possible, this route might lead eventually toward traits that point to artificial consciousness.

Hence, future work could choose to address this and/or more complex “worlds,” which, however, will be very demanding. While this may be a way forward, we would argue that we have presented here a neural-representation-based system that offers a model for Concept emergence within autopoietic systems that only receive information from the environment but not from external supervision processes.

APPENDIX

A. Neural Properties of the System

We use the same activation equations for all neurons in our scheme (except CNN convolutional layers). First, we calculate the weighted sum of the inputs

\[ y = \omega^T u \]

where \( u = (u_1, \ldots, u_n) \) are inputs, \( \omega = (\omega_1, \ldots, \omega_n) \) are weights, \( n = 50 \), and afterward apply sigmoidal saturation

\[ v = f(y) = \frac{1}{1 + e^{-a(y-b)}} \]

where \( a = 0.1 \) and \( b = 100 \).

Activations taken from the CNN can be specified as follows. The network structure, shown in Fig. 3, was trained to distinguish 28 classes: 27 classes are \( 3 \times 3 \times 3 \) feature combinations (size, shape, and color combinations) and one additional output is provided for left versus nonleft position of a shape. This output was added due to insufficient strength of the signal Left in the feature reservoir without this. For network training, we used \( 27 \times 900 \) images of each kind (as described above), which were independently generated for all sets (training or test sets) described above. However, those details are of secondary importance, as we only use this network in our architecture as a feature reservoir and a different network could be used here, too.

The network has the following convolutional layers (interspersed with four max-pooling layers at the bottom of the network): RGB Image \((100 \times 100 \times 3) \rightarrow \text{Layer 1} \) \( (96 \times 96 \times 4) \rightarrow \text{Layer 2} \) \( (44 \times 44 \times 32) \rightarrow \text{Layer 3} \) \( (18 \times 18 \times 64) \rightarrow \text{Layer 4} \) \( (5 \times 5 \times 128) \rightarrow \text{Layer 5} \) \( (3 \times 3 \times 128) \rightarrow \text{Layer 6} \) \((1 \times 1 \times 64)\).

The AAN in our system is supplied with the activations of layers 4–6 (overall 4416 neurons). The convolutional layers have ReLU activation functions, which we normalize to the interval \([0, 1]\). In our study, so that the scale is compatible with the sigmoidal neurons used in the other parts of our system.

Simulated motor activation is triggered in case the activity of all 300 layer 0 neurons in the Big 0 pool summed up exceeds threshold \( F \), hence \( \sum_{j=1}^{300} v_j > F \), with \( N = 300 \) and \( F = 20 \). The same applies to Yellow 0 and Left 0.

B. Learning Mechanisms

In the AAN system, we use four different learning mechanisms, which are all based only on the correlation between inputs. Hence, no explicit supervision is used in the AAN. This is meant to simulate “simple” brains, which have to rely exclusively on signals either from the environment (via their sensors) or arising intrinsically. We use the following.

1) ICO related to heterosynaptic plasticity [62].
2) Conventional Hebbian learning.
3) Above-average Hebbian learning related to mechanisms of synaptic scaling [66], [67], where learning happens only for above-average activity.
4) Above-average Hebbian/anti-Hebbian-balanced learning related to long-term potentiation and depression [78] also coupled to synaptic scaling [66], [67].

Some of these learning rules operate locally on a dendritic branch similar to local learning processes on dendrites in cortical pyramidal cells [79]. Furthermore, we use the well-known physiological fact that weight growth in synapses is limited (large synapses...
grow less \[80\) and implement this through a mechanism that reduces the learning rate when a synapse gets big (annealing of the learning rate).

Note that all here-used learning mechanisms are found in the real neural systems. An in-depth discussion of this would, however, exceed the scope of this article.

We will now first describe the learning rules and afterward explain for which connections they are used and also summarize all layer-specific parameters in the related tables.

1) Input Correlation Learning: This type of learning relies on the correlation between the to-be-learned input correlated with a prewired reflex-inducing input. For example, a big (dangerous) shape triggers a reflex (of running away) by looming over an agent with a bit of a delay relative to the moment the image appears. ICO learning \[62\] makes use of the correlation between these two signals (image input presentation and the looming signal), where a positive correlation leads to weight growth and vice versa. We use a slightly modified version of ICO with

\[
\omega \leftarrow \begin{cases} 
\omega + \mu^{inc}u^{\Phi}, & \text{Reaction required, but not present} \\
\omega - \mu^{inc}u^{\Phi}, & \text{Reaction present, but not required}
\end{cases}
\]

for the synaptic weight vector \(\omega\), with \(\omega(0) = 0\) and input-determined vector \(u^{\Phi} = (u_1^{\Phi}, \ldots, u_n^{\Phi})\), where

\[
u_i^{\Phi} = \begin{cases} 
u_i, & \text{if } u_i > \Phi \\
0, & \text{otherwise}
\end{cases}
\]

Parameters used for ICO rule are provided in Table IV, left. Note that the ICO rule is intrinsically convergent, because learning will stop as soon as the learned synapse is strong enough to supersede the reflex. For proof of this property, see \[81\]. This rule is a typical heterosynaptic plasticity rule, which is common, e.g., in the hippocampus \[82\].

2) Conventional Hebbian Learning: For some of the connections we use regular Hebbian learning given by

\[
\omega \leftarrow \omega + \mu^H(t)uv, \text{ with } \omega(0) = 0.
\]

Stability of learning is assured by an annealing process of the learning rate in case the neuron produces a very high output. This annealing process leads to the final stopping of learning. It is related to the physiological property of limited weight growth, where large synapses do not continue to grow any longer \[80\]. We define

\[
\nu_j(t) > \Theta, \quad \mu_j^H(t) = c\mu_j^H(t - 1)
\]

with \(j = 1, \ldots, N\), where \(N\) is the number neurons in a pool, \(\Theta\) is the threshold, and \(c\) defines the rate of annealing \((c < 1)\). Note that we define \(t = 1, 2, \ldots, t_{\text{max}}\) as the index for the discrete sequence of image presentations. Parameter values are provided in Table IV, right.

3) Above-Average Hebbian Learning: In this type of learning, weights will be strengthened only if inputs and outputs exceed average past activation. The use of average past activation is related to mechanisms of synaptic scaling, where weight growth is determined by the activation level of the neuron \[66\], \[67\]. In our case, this allows bootstrapping, because we start with near-zero weight and, hence, activations are also initially near zero. Thus, without this type of synaptic scaling, learning would not start.

We define

\[
\omega \leftarrow \omega + \mu^{aa}(t)u^{\Delta}H(v^{\Delta}), \text{ with } \omega(0) = 0.01
\]

where \(H\) is the Heaviside function and \(u^{\Delta} = (u_1^{\Delta}, \ldots, u_n^{\Delta})\), where

\[
u_i^{\Delta} = \begin{cases} 
u_i - r_UU_i(t), & \text{if } u_i > r_UU_i(t) \\
0, & \text{otherwise}
\end{cases}
\]

The function \(U\) represents the average past activation and is defined by: \(U_i(0) = 0\) and \(U_i(t) = g_UU_i(t - 1) + (1 - g_U)U_i(t)\) which calculates a sliding average, with \(g_U\) the weighing factor of the averaging history and \(r_U\) the amplitude of the influence of \(U\). The function \(v^{\Delta}\) is defined in the same way as \(u^{\Delta}\).

Hence, learning only takes place if inputs and outputs are strong enough. In addition and different from conventional Hebbian learning, we use the Heaviside function \(7\) and not the actual value of \(v\) for correlation with the input. Hence, learning strength is driven directly by the value of the input \(u\).

Also for this rule, we use annealing of the learning rate, defined as in \(6\) above. All parameter values are given in Table V.

4) Above-Average Hebbian/Anti-Hebbian-Balanced Learning: For this type of learning, we implement weight reduction, related to long-term depression (LTD) in real neurons, to reduce the weights in those inputs which do not support the operation desired by the considered neuron, whereas the weights of the other synapses will grow (long-term potentiation, LTP, \[78\])

\[
\omega \leftarrow \omega + \mu^A(t)u^{\Delta}\text{sign}(v^{\Delta}), \text{ with } \omega(0) = 0.01
\]

Terms \(u^{\Delta}\) and \(v^{\Delta}\) are defined as above. The \text{sign} function takes a similar role as the Heaviside function above, but now weights can grow or shrink dependent on the output.

Also for this rule, we use annealing of the learning rate, defined as in \(6\) above. Parameter values are given in Table VI.

C. Application of the Different Learning Rules and Parameters

The different learning rules have to be used at different target layers to assure correct system behavior.

Feedforward Paths: ICO learning is used for all connections that converge on layer 0 neurons (see Fig. 3). This leads to the first step of separating features from each other but the resulting distributions are still rather dispersed (Fig. 4).

For all other feedforward paths (blue arrows in Fig. 3), we use Above-Average Hebbian Learning. This, in combination with the stopping mechanism, leads to a substantial sharpening of the distributions due to self-organization which leads to a
kind of soft winner-takes-all mechanism favoring all stronger signals.

This type of learning also happens at the feedforward branch of the dendrite of Big 3 (left branch in Fig. 3).

Feedback Paths: The right branch in Big 3 uses balanced (Hebbian/anti-Hebbian) learning (Fig. 3). The utility of this type of learning here is that only neurons obeying IF-THEN rule have the possibility to enhance connections to the branch. Without that the weights would fall to zero.

The final feedback onto the convolutional layers (Fig. 3) uses conventional Hebbian learning, where annealing (6) depends on the maximum output in the pool.

Local Learning at Dual Dendrites: This case concerns the convergence of feedforward with feedback signals at the convolutional layers 4–6 and in the pool Big 3. In general, dendritic branches learn independently by rules, which are different for the feedforward and the feedback branches as described above. This is required to assure that both input types can exert an influence on the output. If input types were pooled for learning, the weaker input would never grow.

Note that the feedforward dendritic branch of the convolutional layers receives activations, which are pretrained and do not change in the course of learning of the AAN network. Local learning on different branches is a common phenomenon found at the dendrites of cortical pyramidal cells [65].

Input Integration Processes at Dual Dendrites: While learning rules are different at the branches, they still use the same output \( v \) for driving their learning rules obtained by using the maximum operation of the saturated sums \( v_1 = f(\omega_1^T u_1) \) and \( v_2 = f(\omega_2^T u_2) \) for the two branches, \( v = \max(v_1, v_2) \). The maximum operator at the junction of the branches is related to well-known gating mechanisms that happen also at dendritic structures [68].

**REFERENCES**

[1] D. Povinelli, “Can comparative psychology crack its toughest nut,” Animal Behav. Cogn., vol. 7, no. 4, pp. 589–652, 2020.

[2] E. Margolis and S. Laurence, “Concepts: The Stanford Encyclopedia of Philosophy,” 2006. [Online]. Available: https://plato.stanford.edu/entries/concepts/ (Accessed: Sep. 15, 2021).

[3] J. A. Fodor, Psychosemantics: The Problem of Meaning in the Philosophy of Mind, vol. 2. Cambridge, MA, USA: MIT Press, 1987.

[4] S. Carey, The origin and evolution of everyday concepts,” in Cognitive Models of Science, vol. 15. Minneapolis, MN, USA: Univ. Minnesota Press, 1992, pp. 89–128.

[5] S. Pinker, The Language Instinct: How the Mind Creates Language. London, U.K.: Penguin, 2003.

[6] A. Atkin, “The Stanford Encyclopedia of Philosophy: Peirce’s Theory of Signs.” 2010. [Online]. Available: https://plato.stanford.edu/entries/peirce-semiotics/ (Accessed: Sep. 15, 2021).

[7] F. G. Varela, H. R. Maturana, and R. Uribe, “Autoeposies: The organization of living systems, its characterization and a model,” Biosystems, vol. 5, no. 4, pp. 187–196, 1974.

[8] J. Von Uexkull, A Stroll Through the Worlds of Animals and Men: A Picture Book of Invisible Worlds. New York, NY, USA: Walter de Gruyter, 1992.

[9] A. Sloman, “Requirements for a Fully-Deliberative Architecture (or Component of an Architecture).” 2014. [Online]. Available: http://www.ual.ac.uk/research/projects/cogaff/misc/fully-deliberative.pdf

[10] M. Minsky, “Steps toward artificial intelligence,” Proc. IRE, vol. 49, no. 1, pp. 8–30, 1961.

[11] P. H. Winston, “Learning structural descriptions from examples,” Ph.D. dissertation, Dept. Electr. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 1970.

[12] H. P. Nii, E. A. Feigenbaum, and J. J. Anton, “Signal-to-symbol transformation: HASP/SIAP case study,” AI Mag., vol. 3, no. 2, p. 23, 1982.

[13] S. Harnad, “The symbol grounding problem,” Physica D Nonlinear Phenomena, vol. 42, nos. 1–3, pp. 335–346, 1990.

[14] S. Harnad, “Symbol grounding problem,” Scholarch, vol. 2, no. 7, p. 2373, 2007.

[15] S. Coradeschi, A. Louiti, and B. Wrede, “A short review of symbol grounding in robotic and intelligent systems,” Künstliche Intelligenz, vol. 27, no. 2, pp. 129–136, 2013.

[16] E. Gat, R. P. Bonnasso, and R. Murphy, “On three-layer architectures,” in Artificial Intelligence and Mobile Robots, vol. 195. Menlo Park, CA, USA: AAAI Press, 1998, p. 210.

[17] D. C. Dennett, Consciousness Explained. London, U.K.: Penguin, 1993.

[18] C. Yu and D. H. Ballard, “On the integration of grounding language and learning objects,” in Proc. AAAI, vol. 4, 2004, p. 488–493.

[19] T. Nakamura, T. Nagai, and N. Iwahashi, “Grounding of word meanings in multimodal concepts using LDA,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2009, pp. 3943–3948.

[20] A. Cangelosi, “Grounding language in action and perception: From cognitive agents to humanoid robots,” Phys. Life Rev., vol. 7, no. 2, pp. 139–151, 2010.

[21] A. Aly and T. Taniguchi, “Towards understanding object-directed actions: A generative model for grounding syntactic categories of speech through visual perception,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2018, pp. 7143–7150.

[22] L. El Hafi, H. Nakamura, A. Taniguchi, Y. Hagiwara, and T. Taniguchi, “Teaching system for multimodal object categorization by human-robot interaction in mixed reality,” in Proc. IEEE/RSJ Int. Symp. Syst. Integ. Robots (SII), 2021, pp. 320–324.

[23] S. Coradeschi and A. Saffiotti, “An introduction to the anchoring problem,” Robot. Auton. Syst., vol. 43, nos. 2–3, pp. 85–96, 2003.

[24] P. Vogt, “Anchoring of semiotic symbols,” Robot. Auton. Syst., vol. 43, nos. 2–3, pp. 109–120, 2003.

[25] M. Wächter et al., “Integrating multi-purpose natural language understanding, robot’s memory, and symbolic planning for task execution in humanoid robots,” Robot. Auton. Syst., vol. 99, pp. 148–165, Jan. 2018.

[26] M. J. Aein, E. E. Aksoy, and F. Wörgötter, “Library of actions: Components based on simple language instructions,” AI Mag., vol. 38, no. 2, pp. 140–154, Jun. 2019.

[27] F. Wörgötter et al., “Structural bootstrapping—A novel, generative mechanism for faster and more efficient acquisition of action-knowledge,” IEEE Trans. Autom. Mental Develop., vol. 7, no. 2, pp. 140–154, Jun. 2015.

[28] M. Tamosiunaite et al., “Cut & Recombine: Reuse of robot action components based on simple language instructions,” Int. J. Robot. Res., vol. 38, nos. 10–11, pp. 1179–1207, 2019.

**TABLE V**

| Parameters for Above-Average HEBBIAN Learning. Numbers 1, 2, and 3 Refer to the Layer Indices, NOT THAT Layer 1 NEEDS Slower Annealing, as Its Inputs Are Wider Distributed (See Histograms of Layer 0 in Fig. 4) Than in the Other Layers |
|---|---|---|---|---|---|---|---|
| \( \mu(0) \) | \( \mu(0) \) | \( \mu(0) \) | \( U \) | \( V \) | Annealing, \( c \) |
| \( 1 \) | \( 2 \) | \( 3 \) | \( AND 1 \) | \( AND 2 \) | \( AND 1 \) | \( AND 2 \) | \( AND 1 \) | \( AND 2 \) | \( AND 1 \) | \( AND 2 \) |
| 5 | 0.05 | 0.025 | 0.9 | 1.0 | 0.9 | 0.9 | 0.99 | 0.9 | 0.99 | 0.9 | 0.99 | 0.095 | 0.1 |

**TABLE VI**

| Balanced | Hebb | \( \mu(0) \) | \( U \) | \( V \) | Annealing |
|---|---|---|---|---|---|
| \( \omega_1 \) | \( \omega_2 \) | \( \omega_1 \) | \( \omega_2 \) | \( \omega_1 \) | \( \omega_2 \) |
| 0.05 | 0.9 | 1.0 | 0.9 | 1.0 | 0.995 | 0.1 |
