HAL
open science

Towards Deep Developmental Learning

Olivier Sigaud, Alain Droniou

To cite this version:

Olivier Sigaud, Alain Droniou. Towards Deep Developmental Learning. IEEE Transactions on Cognitive and Developmental Systems, 2016, 8 (2), pp.99-114. 10.1109/TAMD.2015.2496248 . hal-01331799

HAL Id: hal-01331799

https://hal.sorbonne-universite.fr/hal-01331799v1

Submitted on 14 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Deep Developmental Learning

Olivier Sigaud, Alain Droniou

***Abstract*—Deep learning techniques are having an undeniable impact on general pattern recognition issues. In this paper, from a developmental robotics perspective, we scrutinize deep learning techniques under the light of their capability to construct a hierarchy of meaningful multimodal representations from the raw sensors of robots. These investigations reveal the differences between the methodological constraints of pattern recognition and those of developmental robotics. In particular, we outline the necessity to rely on unsupervised rather than supervised learning methods and we highlight the need for progress towards the implementation of hierarchical predictive processing capabilities. Based on these new tools, we outline the emergence of a new domain that we call *deep developmental learning*.**

***Index Terms*—deep learning, developmental robotics, sensorimotor contingencies, affordances, hierarchical predictive processing**

I. Introduction

Consider a mug on a table. You can recognize it from any perspective. You distinguish it more or less quickly from other mugs, from a cup, from a glass, etc., though you also immediately perceive similarities to these other objects. You would probably recognize it in the dark just by touching it with your fingers. You know how it feels in your hand and how heavy it is. You know the proper way to handle it, you can grasp it in many ways for different purposes (move it somewhere else, give it to someone else, put it upside down or drink from it). You know what it is designed for, you know that you can pour some liquid into it, but you also know that you could put it on some paper before a gust of wind blows the paper away. You know whether it would break if you threw it down. You know that it is an everyday object, you know that the name of this object is "mug", you know where you can purchase another one, etc. And you still know much more about this mug, more than could be said in a research paper.

Endowing an artificial system with this kind of human-like knowledge is the central concern of Artificial Intelligence (AI), a discipline that has initially focused in modeling abstract knowledge at the "symbol level" [1]. However, designing *artificial symbol systems* that were processing some data into computers was soon criticized through Searle's Chinese Room argument [2]. Searle convincingly argued that symbols processed in this way were meaningless for the computer, because they were essentially disconnected from the world. This critique was reformulated as the "Symbol Grounding Problem" [3] who pointed that the symbols manipulated by these *disembodied* AI systems should be *grounded* into some sensorimotor interaction with the world. Actually, a good deal of the kind of rich knowledge listed above is about perception and about action. So, if we want an artificial system to use such kind of knowledge, this system must perceive and act from its own perspective in the world, it must have several sensors giving access to different modalities, it must be endowed with interaction capabilities, in a word it must be a *robot* [4], [5].

With the standard AI methods and tools, one could already design experiments where a robot which is asked to bring a coffee builds a plan, finds a mug in its immediate environment, pours some coffee in it and brings it to the user. This can be achieved by designing ontologies and symbolic rules such as "to bring coffee to the user, pour it in a mug". The elementary entities of these ontologies and rules, such as "mug", "coffee" and "user", are represented by simple tokens or symbols, that must be recognized through dedicated perception modules using predetermined sets of visual features. For elementary actions like "pour", the engineer must build dedicated motor programs. In a way, these approaches solve a weak version of the symbol grounding problem. Indeed, the symbol "mug" and other symbols are connected to what is perceived in the environment through the *ad hoc* vision and motor modules.

However, a key aspect of our intelligence that is overlooked by such attempts is that the richness of our sensorimotor and sub-symbolic knowledge about everyday objects plays a central role in the flexibility and adaptivity of our behavior. Extending the above approach to better account for this rich knowledge would require a larger engineering effort to build a richer ontology and more robust perception and actions capabilities. Though impressive results have been obtained with this line of research, e.g. [6], engineering in this way all the behaviors that would be required to display general human level intelligence is out of reach.

From this perspective, the point is that most of our rich sensorimotor knowledge about an object has been acquired through direct physical interaction with this object and with others, thus this knowledge is *learned* and essentially depends on our history of interaction with the world. Developmental robotics is a principled approach to AI in robotics that relies on this evidence. Its main endeavor consists in endowing a robot with the capability to build by itself the rich knowledge about everyday objects described above and to use this knowledge appropriately in any context.

The key idea is that, instead of engineering the perception and action capabilities themselves, roboticists should design the necessary learning mechanisms to let the robot build its own perceptive and behavioral repertoire, through its interactions with its physical and social environment. As a result, the developmental approach to intelligence strives to endow robots with all the learning capabilities that may be necessary to build rich and flexible sensorimotor representations from the sensorimotor to the symbolic level [7], [8], [9].

The authors are with Sorbonne Universités, UPMC Univ Paris 06, UMR 7222, F-75005 Paris, France, CNRS, Institut des Systèmes Intelligents et de Robotique UMR7222, Paris, France olivier.sigaud@isir.upmc.fr +33 (0) 1 44 27 88 53

One source of information about the learning capabilities that are required to build knowledge from interaction with the world is developmental psychology [10], [11], [12]. By observing child development, one may understand the necessary conditions for an artificial system to build such knowledge from its basic interaction capabilities [13], [14], [15].

A psychological notion that is not intrinsically developmental but plays an important role in developmental robotics is the notion of *affordance* [16]. Quickly stated, affordances of an object for an agent are actions that this agent immediately recognizes as applicable to this object, together with some knowledge of the effect of these actions, but without necessarily identifying the object itself.

Affordances are important because they provide a solid ground for interacting with objects to solve everyday problems. In particular, their focus on the effect of actions makes them appropriate for planning or higher level reasoning about interactions with the corresponding objects. For instance, if I see a sheet of paper about to be blown away and if I know that putting a mug-like object on a paper prevents it from being blown away, then I can decide to put the mug on the paper, without even recognizing it as a mug. Based on this view of affordances, many researchers address higher level issues, such as the capability to generalize affordances across similar objects, learning inter-objects affordances [17], or solving the bootstrapping problem, where lower level affordances should be used by the architecture to build higher level affordances, without giving this hierarchical structure in advance [17], [18].

With this new perspective, learning affordances replaces the endless effort of building ontologies and writing rules while endowing the robot with similar problem solving capabilities. This may lead to a large reduction of the aforementioned engineering effort. So, is the problem solved? Clearly not: The general capability to manipulate unforeseen objects and tools has not yet been demonstrated on a robot or in simulation with this approach. Why is this? There are two issues.

First, the developmental route to autonomous robots being long and difficult, there is a strong temptation to use shortcuts by engineering into the robot some of the competences that it should discover by itself. This temptation is particularly obvious at the lowest levels of sensory processing. Indeed, learning affordances is understood mainly as learning the effects of actions on objects. As a first approach, this can be implemented by defining an effect space with a set of dedicated features and clustering this space into a discrete set of effects. For instance, by using a "speed of movement" feature, one can easily determine that some objects move when they are pushed and some others do not. This approach is taken in many state-of-the-art research works [19], [20], [21]. This is very clearly illustrated for instance in [22], where the first box in the architecture is labeled "compute basic features" and outputs sizes, shapes, etc. With this perspective, the engineer's expertise is required twice: to determine the features of the effect space and to provide dedicated perceptual modules that acquire the value of these features. As a consequence, most proof-of-concept experiments in affordance-based developmental robotics seem to suffer from similar restrictions as the ontology-based ones.

The second issue is deeper. The point is that affordances themselves are often represented as abstract symbolic entities in higher level processes such as planning and reasoning. According to the *embodied cognition* perspective, this is unsatisfactory because higher level cognition is not just a matter of manipulating abstract symbols. According to the embodied cognition perspective, "conceptual representations that constitute our knowledge are grounded in sensory and motor experiences, and processed at this sensorimotor level, rather than being represented and processed abstractly in an amodal conceptual system" [23]. If we take this perspective seriously, solving the symbol grounding problem is not only a matter of learning the mapping from the sensorimotor flow to a level of abstract symbols. The conceptual representations we manipulate for planning or reasoning being modal, not to say multimodal, there should be no modal-to-abstract discontinuity between the lowest level sensorimotor information processing modules and those which manipulate higher level representations.

From these insights, it appears that the crucial element that is missing to build the kind of rich sensorimotor knowledge we have about objects is a flexible, general purpose mechanism starting from the multimodal sensorimotor flow of the robot and building the embodied, sub-symbolic representations that are necessary to predict, reason and plan at all levels of our cognitive capabilities so as to finally solve the symbol grounding problem.

The primary goal of this paper is to investigate whether the so-called *deep neural networks* and the corresponding *deep learning* techniques may provide such mechanisms and representations. Deep learning is a fast growing trend in machine learning that has attracted a lot of attention by outperforming all other techniques in visual, auditory, speech and text pattern recognition tasks [24], [25]. It consists in training neural networks with several layers in such a way that the input data is accurately encoded by building more and more complex features as the activation signal goes higher and higher into the layers. Since they start from raw sensory signal and flexibly learn relevant features by themselves, the basic intuition is that deep neural networks might provide at all levels the adequate representations of the rich knowledge developmental robotics should be looking for.

However, the methodological constraints of pattern recognition differ from those of developmental robotics. In particular, a supervised learning step is generally used in pattern recognition, whereas labeling the data might turn out to be difficult in developmental robotics. Thus, even if the perception of robots can readily be improved using the current state-of-the-art deep learning techniques, further requirements should be pursued to address the development of hierarchical representations in robots with these techniques. As a consequence, a secondary goal of this paper is to identify in the deep learning literature the tools that are the most adequate under the developmental robotics constraints, giving rise to the emergence of a new domain built on these tools that we call *deep developmental learning*.

Less technically, this paper also provides an opportunity to revisit central concepts of developmental robotics with

new conceptual tools, which may result in a new general perspective on the development of cognition, of interest not only to developmental roboticists, but also to all researchers interested in computational models of cognitive development, e.g. [26], [27], [28].

The paper is organized as follows. In Section II, we give a broad overview of deep neural networks and the related learning techniques. In Section III, we highlight the immediate respects in which deep learning techniques may help circumvent the limitations of standard approaches to perception in developmental robotics. We show that they provide the adequate tools for building flexible and hierarchical representations, starting from the raw multimodal information a robot gets from its sensors and building towards higher level representations that are needed to adequately interact with the world. In Section IV, we examine limitations of standard deep learning research when developmental robotics applications are considered. First, we highlight the necessity to rely on unsupervised rather than supervised learning methods and describe preliminary attempts in this direction. Second, we advocate for a stronger focus on methods able to predict temporal information on-line and to deal with hierarchical predictive processing and temporal chunking issues. In Section V, we also reconsider issues such as behavior optimization, action selection, curiosity or life-long learning, under the light of deep learning techniques. Finally, as a conclusion, we outline the need for more computational power for embedding deep learning in robots, we sketch the role of "dreams" under this perspective, and we outline the long way that still has to be travelled before true human-like symbolic intelligence issues can be addressed.

II. Deep neural networks in a nutshell

A neural network implements a generally non-linear function between some input and output data through weights connecting computational units called *neurons*. It is used as a flexible representation for capturing regularities between the input and the output. When the connection graph between neurons contains cycles, the neural network is said *recurrent* and can capture dynamical regularities over multiple times steps. Otherwise, it is said *feed-forward* and is generally organized into layers. It is well known that, even with a single hidden layer, a neural network can approximate any continuous function with arbitrary precision [29]. However, when the input dimensionality increases, when the function becomes more irregular or when the desired precision increases, the number of required hidden units grows exponentially. A nice property of deep neural networks comes from the fact that using multiple levels of hidden layers decreases the number of units required for representing a large set of functions (from exponential to linear complexity [30]) by factorizing representations.

Among early attempts to benefit from this property, some deep neural network architectures dedicated to pattern recognition in images obtained very good performance [31]. These architectures were inspired from the structure of the visual cortex and could be trained with a standard error back-propagation algorithm. However, applying this algorithm to more general deep architectures proved difficult. This is due to the so-called "vanishing or exploding gradient problem" [32], [33], [34], resulting from the fact that these networks easily reach a solution domain where the gradient goes to zero or diverges.

To overcome this issue, a breakthrough consisted in training each layer in sequence to "reconstruct" its own input as output [35]. This unsupervised training process, also called "pre-training" benefits from the immediate availability of a *reconstruction error* which is a distance between the output and the input. Once a layer correctly reconstructs its input, another layer can be added or *stacked* on top of it and pre-trained to do the same with the hidden layer or the reconstructed output of the previous layer as input. Once the overall architecture is trained in this way, it can be fine-tuned for achieving a more specific task. This latter *fine-tuning* stage can be done according to either a similar reconstruction error cost on the input data set or a task specific cost function (e.g. [36]). Another possibility is to stack a classical supervised algorithm, e.g. a Support Vector Machine [37] on top of the network, using as input the activity of the top layer of the deep network [38].

Pre-training proved efficient at initializing the weights of the network to avoid getting stuck in flat gradient landscapes, leading to a surge of interest in networks with many layers [39]. In the following years, the resulting deep neural networks outperformed state-of-the-art results in many pattern recognition tasks [40], [41], [42], which made these methods extremely popular [25].

There are many general reviews on deep neural networks [24], [25]. However, the domain is evolving very fast and questions so fundamental as why pre-training techniques work well in practice are still incompletely understood. As a consequence, this is the matter of intensive research and debates between several competing explanations [43], [44], [45]. Furthermore, the general assumption that training each layer with an input reconstruction stage plays a key role in the performance of deep neural networks is itself questioned e.g. by [46] who obtained a similar performance without this input reconstruction stage using a second order gradient (Hessian-free) technique. Other works also suggest that given enough time and computational power, it is possible to outperform pre-trained networks with a standard, carefully tuned back-propagation algorithm applied to the whole network [47], [48].

Thus, in this section, rather than striving to give a technical account of the domain that would be outdated even before publication, we only briefly describe four families of deep neural networks, namely Convolutional Neural Networks [31], Auto-Encoders [49], Restricted Boltzmann Machines [50] and Gated Networks [51], that can be used as building blocks or starting points for many state-of-the-art deep neural networks architectures.

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are an early family of deep learning architectures inspired from the human vision system that proved particularly efficient for image processing

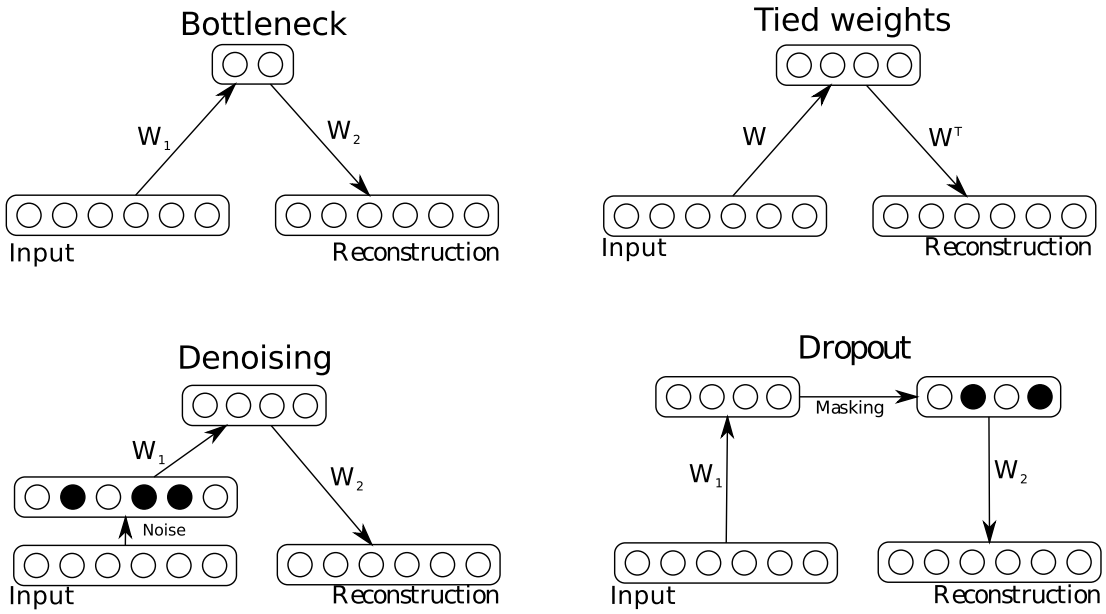

Fig. 1. Most popular regularization methods for auto-encoders. The bottleneck technique consists in encoding data over a small number of neurons. The weight tying technique consists in using the same weight matrix for encoding and decoding. For the denoising technique, data are first corrupted (for instance by setting some of the input neurons to 0), before being encoded and decoded. The dropout technique, finally, consists in setting to 0 a portion of hidden neurons before decoding.

applications [31]. As clearly illustrated in Fig. 2 of [52], they are generally composed of 6 or 7 layers of neurons, where filtering or convolutional layers alternate with feature map or pooling layers. A surprising and still unexplained fact is that these quite deep architectures do not seem to require an unsupervised training step to perform well, in contrast with many other deep architectures [24]. A crucial aspect for the perspective of this paper is that these architectures generally require some labeled data to perform supervised learning, which is inadequate from a developmental robotics perspective. However, there are some exceptions, such as [53].

B. Auto-Encoders

Auto-Encoders are the simplest deep architectures that can be pre-trained with unsupervised learning. To learn an accurate representation of their input, they are trained to minimize the reconstruction error of input data by back-propagating it from the output layer to hidden layers. However, if the goal was just to reproduce the input as output, the simplest model would be the identity function. In order to prevent this from happening and get representation with good generalization capabilities, several regularization techniques have been proposed, among which (see Fig. 1):

- The bottleneck approach [45] consists in enforcing *dimensionality reduction* into the hidden layer by having fewer neurons in the hidden layer than in the input layer.
- Denoising auto-encoders (DAEs) [54]: the input is first corrupted with noise (e.g. randomly setting some input units to zero), and the reconstruction error is measured either compared to the non-corrupted input, or to the same input corrupted with independent noise.
- Dropout [55], [56]: the hidden layer is corrupted with noise (here again, randomly setting some units to zero, often about 50% of them), which constrains the different neurons to learn independent features. The noise is removed after training and some renormalization is necessary to take into account the fact that the signal is better propagated when noise is removed.
- Weight tying: weights for encoding and decoding are tied. Actually, this last mechanism does not prevent one from using the identity function as model, so it should be used in conjunction with another mechanism.

Two other techniques are not represented in Fig. 1:

- Contractive auto-encoders (CAEs) [57]: a penalty cost is added to penalize the Jacobian of the hidden layer with respect to the input. This aims to contract the learned representation along the relevant dimensions to represent the input. Higher-order contractive auto-encoders [58] also penalize higher-order derivatives.
- Sparse auto-encoders [59]: a sparsity constraint is added to the hidden layer activity so that most units stay inactive.

C. Restricted Boltzmann Machines

Restricted Boltzmann Machines (RBMs) [60] are a subclass of Boltzmann machines [61] which are themselves a specific instance of Probabilistic Graphical Models [62]. They consist of two layers, the visible layer corresponding to the input, and the hidden layer corresponding to an internal representation of the underlying source of the input data. At first glance, the connection between RBMs and auto-encoders is weak. While auto-encoders are deterministic models trained to reconstruct their input as output, RBMs are stochastic models whose function is to model the distribution of their input data through the activity of their hidden layer, which is usually binary

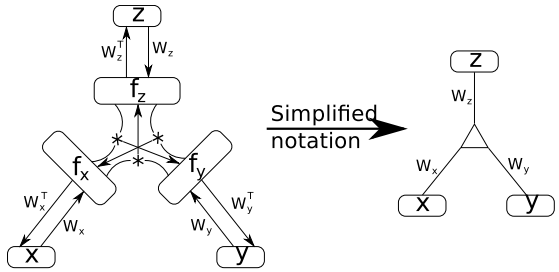

Fig. 2. General view of a gated auto-encoder and simplified notation. In the bottom-up use of the network, given two input $\mathbf{x}$ and $\mathbf{y}$, their projections through W_x and W_y are multiplied in an intermediate *factors* layer, before being projected on the hidden layer $\mathbf{z}$. Such connections are useful to learn representation of multi-dimensional relations between different variables. The role of the three external layers can be exchanged, making it possible to compute a reconstruction of $\mathbf{y}$ given $\mathbf{x}$ and $\mathbf{z}$, and *vice versa*: the two input are projected on the factors layer, and the resulting factors are projected on the third layer. The simplified notation is reused in Fig. 3 and 4.

(though real-valued hidden layers are possible, see e.g. [63]). As such, RBMs are *generative* models (see [26] for a tutorial introduction). This is in contrast with CNNs and standard auto-encoders which are only *discriminative*, i.e. the information flow is oriented from the input to the output, thus they can only infer the output corresponding to an input [52].

However, a deeper analysis of RBMs and auto-encoders under the light of *energy-based models* (see [64], [65] for a technical presentation) reveals that, under some conditions, they are not so dissimilar. In particular, DAEs can be analyzed as generative, stochastic models which locally characterize the distribution over input data. Those models received an increasing interest recently because, being deterministic, they benefit from a simpler training procedure than RBMs [66], [67].

The same regularization techniques used for auto-encoders can also be used for RBMs (bottleneck, dropout, etc.). Nevertheless, the need for regularization is weaker in RBMs, because the random sampling of neuron activities in RBMs exerts a strong regularization constraint. Actually, they have been shown to have good generalization capabilities even with high dimensional inputs [68].

Finally, RBMs themselves are not deep networks, but they are the key building block of Deep Belief Networks (DBNs), a popular family of deep neural networks which are usually trained layer per layer in an unsupervised way, exactly as for auto-encoders.

D. Gated Networks

A more recent family of generative models is the *factored* and/or *gated* networks family, that connect three or more layers together instead of two. Gated RBMs were introduced in [69] whereas Gated Auto-Encoders (GAEs), illustrated in Fig. 2, were proposed in [70].

The general idea behind these networks is that the relationship between two layers is multiplicatively gated by the activity of the third layer into an intermediate *factor* layer. The network shown in Fig. 2 can combine bottom-up and top-down interactions corresponding to different ways to use it. First, given the lower left layer (labeled "x") and the lower right layer (labeled "y"), the network can infer the higher level layer (labeled "z") in a bottom-up way. Second, given the "x" layer and some top-down information about the "z" layer, the "y" layer can be predicted. Several networks using GAEs are introduced later in this paper. See [71] for a more detailed presentation.

III. Impact of deep learning on developmental robotics issues

As put forward in the introduction, developmental robotics is missing a representational framework for sub-symbolic knowledge and the mechanisms to learn it from the sensorimotor information a robot receives. In this section, we examine what deep learning techniques can immediately bring to address this issue. We highlight that they are adequate to start from raw sensorimotor information and build hierarchical and relevant representations while dealing with the curse of dimensionality and combining bottom-up and top-down information processes. Additionally, the outstanding pattern recognition capabilities of deep learning techniques can provide adequate tools for developmental robotics in more specific sub-domains such as natural language related processes and human robot interaction.

A. Starting from raw sensorimotor information

To avoid engineering features as outlined in the introduction, the representations learned by a robot should be built directly from the multimodal flow of information it receives from all its sensors (e.g. visual, auditory, tactile, proprioceptive, etc.), i.e. the raw *sensations* of the robot [72]. The structure of these sensations obey certain rules that are called *sensorimotor contingencies* and that may help us recognize the different modalities we experience [72].

Some of these rules are independent of the objects the agent is interacting with, such as the fact that moving forward generates a visual flow expanding from the center of vision. Such rules are characteristic of modal sensations. Other rules are characteristic of the experience of *perceptions*, when they are related to some perceptive attributes of objects. For instance, when touching an object, the specific tactile feedback we experience depends on the object itself but also on our way of touching. This tactile feedback can be interpreted as the *effect* of touching this object in this way. It is often the case that sensorimotor contingencies correspond to regularities in the coupling between sensations or perceptions and actions. Thus, from many examples given in [72], it is clear that actions in the world such as head movements are necessary to turn raw sensations into a perception, because these actions reveal the underlying structure of the coupling between sensations and actions, i.e. the sensorimotor contingencies. Moreover, from this perspective, the perception or recognition of the object can be considered as the effect of acting on this object. Thus learning sensorimotor contingencies accounts at a very elementary level for a phenomenon that is foundational to learning affordances, i.e. the recognition of the object through consequences of our actions on it.

The fact that our sensations and perceptions obey certain rules defines sensorimotor contingencies as regularities over

these domains, or as *manifolds* where sensory or perceptive data is concentrated into the underlying. A key feature of deep learning techniques is that they are good at extracting such manifolds, and they can be applied to large spaces such as the sensory space of a robot, thus they look adequate for extracting *sensorimotor contingencies* into the first layers of a hierarchy of representations built by a robot [73], [74].

B. Building sensorimotor representations that make sense

In a deep architecture, each layer is extracting a set of features to encode the information provided by the previous layer. By stacking a set of layers onto each other, deep neural networks naturally extract features hierarchically from the input signal. Actually, it has been shown that the features extracted in such a way generally make sense to the external observer. For instance, a network trained on a data set containing pictures of faces, airplanes, chairs and bikes learned Gabor-like features at the lowest level, which were progressively combined into more complex features, such as eyes, wings and wheels in intermediate layers, towards global prototypes of each class in the top layer [53]. Other examples can be found in natural language processing, speech processing or machine translation, where the study of the encoding performed by deep neural networks reveals that related concepts are classified in a meaningful way (see e.g. [75]). There are now many techniques to extract different views of the representations a deep neural network has learned [26].

Thus, deep learning techniques are endowed with the right properties for the implementation of hierarchical sensorimotor representations, since they extract the adequate features from raw data to build a hierarchy of useful sub-symbolic representations that make sense for the external observer and for higher level processing modules.

C. Dealing with the curse of dimensionality

If we want a robot to build its own representations from raw sensations rather than from features dedicated to a specific task, the corresponding learning techniques should be able to face the *curse of dimensionality* [76], because the raw sensory space of any human-like robot is potentially huge.

Starting from raw input, deep neural networks are able to reduce the dimensionality of data by representing only the relevant manifolds. A such, they offer the adequate premises for dealing with raw sensors, because they can address very high dimensional input spaces, provided enough computational power.

Moreover, the hierarchical nature of representations plays a key role in dimensionality reduction because it allows the factorization, reuse and transfer of knowledge at all levels and in various domains (e.g. the concept of ”eye” can be learned from human faces and transferred to dogs, cats, etc.) [39], exactly as we do [77]. This is also a key to knowledge transfer from one domain to another. Finally, the hierarchical representations built in deep neural networks progressively shrink the input dimensionality while carrying more and more semantic representations, as already outlined in Section III-B.

D. Building multimodal representations

Multimodal information processing has been recognized as a central concern in developmental psychology and developmental robotics [78]. Actually, it is well-known that, for interpreting the continuous flow of information arising from our sensors, we are using both monomodal correlations, i.e. correlations between informations coming from the same sensor, and multimodal correlations, i.e. correlated information coming from different sensory modalities, such as touch, vision or audition. We generally benefit from multimodality to interpret our current situation, but multimodal processing can also be revealed by maladaptive effects that it generates. For instance, in the McGurck effect [79], the perception of a syllable differs depending on the presence of visual only, auditory only, or both visual and auditory stimulus. The rubber hand effect [80] is another example, where a coupling between visual and tactile stimuli modifies the perceived body. These examples show that multimodal fusion is not only a high level process, but that different modalities are processed in an intertwined manner even at the lowest levels of perception. An advantage of using a unified low level sensorimotor representation is that it facilitates this multimodal integration.

In that respect, an important property of deep neural networks is that their internal processes are agnostic about the nature of the input data. Thus, the raw data coming from very different sensors can be treated with the same mechanisms and merged into higher level representations. As a consequence, these networks can be very naturally endowed with multimodal information processing capabilities. There exist several approaches to multimodality in deep learning and developmental robotics. We refer the reader to [81] for a survey of these works.

E. Combining bottom-up and top-down information processes

Combining bottom-up and top-down information processes is mandatory from the developmental robotics standpoint. In one way, high level signals such as natural language have a top-down influence on the structuring of our low level representations of the world. But, in the other way, the bottom-up structuring of our sensorimotor representations plays a key role in the acquisition of referential tools such as language. More generally, bottom-up and top-down information processes influence each other and their interplay is crucial for the structuration of our representations and the development of higher level processing capabilities [82].

Thus, it is crucial from the developmental robotics perspective that some of the standard generative approaches to deep learning such as RBMs and GAEs readily combine bottom-up and top-down interactions. As generative models, RBMs can combine bottom-up and top-down information processes because their learning and inference processes involve a flow of information from the visible to the hidden layer and *vice versa* [26]. In Section IV, our focus is more on GAEs than on RBMs, because these tri-partite networks allow to combine the bottom-up and top-down flow with a third direction, namely the temporal flow of information.

F. Natural language related and socially oriented learning

A learning robot should build representations of its interactions with its physical environment, but more socially oriented learning processes should also be considered.

Many of these processes are related to natural language. In developmental robotics research, a lot of effort is dedicated to language acquisition, speech and text understanding, the development of writing capabilities and other language related processes [7]. Deep learning techniques have shown their efficiency at dealing with many of these language related issues [83], [84]. Thus, they should naturally contribute to the developmental robotics research effort dedicated to them. However, they should do so within a conceptual framework that is compatible with the requirements of developmental robotics, rather than based on supervised learning methods and labeled data, as is mostly the case in this literature. Furthermore, such processes should be integrated into the more global concern of building high level representations of interactions with the world, since they play a key role in the top-down restructuring of these representations.

A closely related concern where deep learning techniques can have a large impact is human-robot interaction. Learning to interact with a caregiver is fundamental for the developmental process of infants. This issue is being taken more and more seriously in developmental robotics. Some of these interactions are conveyed through natural language, but others are more related to non-verbal signals, such as posture or gesture, etc. Here again, a lot of effort is dedicated to these concerns in the deep learning literature, with the same methodological limitations as above [85], [86].

G. Summary

To conclude this section, a nice thing about deep learning techniques is that they have the right properties for learning sensorimotor contingencies from the raw multimodal and sensorimotor signal and building hierarchical representation of higher level regularities from this flow. Actually, extracting hierarchical regularities in large spaces is exactly what deep neural networks are good at. Additionally, more specific pattern recognition capabilities in image, speech or human gesture recognition can be beneficial to developmental robotics research by providing concrete means to improve the interaction and learning capabilities of robots.

Thus, a first interest of deep learning for developmental robotics is that the appealing properties of deep neural networks can readily be imported into the design of perception and representational systems for robots, resulting in an immediate improvement in the flexibility of their sensory representations. However, a *deeper* perspective is possible, that we present in the next section.

IV. Better tools for deep developmental robotics

In the previous section, we outlined what the outstanding pattern recognition capabilities of deep learning techniques can bring to developmental robotics. However, while doing so, we overlooked two important limitations of most of the deep learning techniques reviewed so far.

First, they generally call upon some supervised learning step. Using supervised learning is legitimate in domains like text or speech processing where a lot of labeled data is available. By contrast, it is much more questionable in the developmental robotics context where labeling the sensorimotor experience of an autonomous robot is hardly feasible.

Second, deep learning techniques generally focus on static patterns, whereas robots are intrinsically engaged in dynamic interactions with their environment. One may build robotics architectures that take these static patterns as input, but this approach would prevent cognitive operators that deal with these patterns to emerge from the information processes themselves. Our deeper view of developmental robotics is that the cognitive processes and operators themselves should be learned into the same representational substrate as the manipulated representations, which implies that such representations are themselves dynamical.

In this section, we investigate two complementary lines of research dedicated to addressing both limitations above. In Section IV-A, we describe a recent effort to learn symbolic representations from high dimensional data by relying exclusively on unsupervised learning mechanisms. We show that the corresponding work addresses many of the requirements listed in the previous section, though it does not address all developmental robotics constraints. In Section IV-B, we survey various attempts to design deep architectures that can deal hierarchically with temporal information in order to predict what will happen next, as we consider this as crucial for deep developmental robotics.

A. Learning symbols from high dimensional data

As already outlined in Section III-C, starting from the raw sensorimotor data of a robot to build hierarchical representations raises a major dimensionality problem. The work presented in [81] can be seen as a direct attempt to address it while combining two complementary requirements: perform dimensionality reduction by learning along meaningful manifolds, and rely exclusively on unsupervised learning methods.

As for the first requirement, an important assumption in pattern recognition is that input data is generally concentrated in the vicinity of non-linear sub-manifolds of much lower dimensionality than the input space. Actually, the higher the dimensionality of the input space, the most likely this assumption [87], [88], [57]. Thus, rather than performing dimensionality reduction by projecting input data into a lower dimensional space, one may tune the parameters of the representation so that it preferably represents areas in the original space where the data density is high [89], [90]. This can be achieved by applying a contractive operator that concentrates the output where the input data lies [49].

As for the second requirement, the critical point of most standard unsupervised learning algorithms is the definition of a suitable metric that is used to cluster the samples according to their relative distance. Most of these algorithms [91] are variants of the K-MEANS [92], hierarchical clustering [93] or Kohonen-like networks [94]. They all rely explicitly on the complex definition of a metric that usually depends on the addressed task.

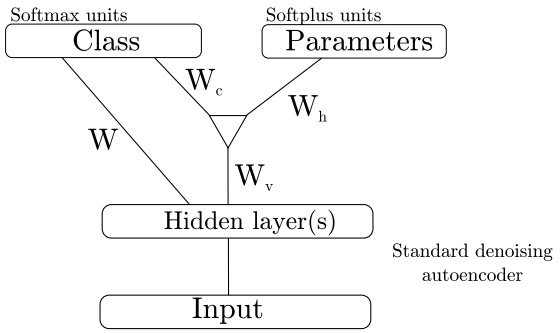

Fig. 3. Gated network for unsupervised classification (adapted from [81]). The network aims at clustering input data at the softmax layer using the W matrix. Based on this clustering, a gated network learns to represent the underlying manifold. Gated connections make it possible to share features W_{v} and W_{h} between all classes and reduce the number of parameters to learn. The hidden layer(s) are intended to implement a standard deep neural network to handle high-dimensional, raw perception as input.

An important step towards removing the necessity of supervision in deep neural networks was achieved by sharing features in the gated architecture described in Section II-D [69], [51]. The presence or absence of each learned feature was indicated by Boolean variables in a hidden layer. For k hidden units, the authors demonstrated that their model was equivalent to a mixture of 2^k logistic classifiers with shared weights. They reported that sharing features improved the classification performance of the system compared to an unsharing approach.

The architecture illustrated in Fig. 3 was an extension to the factored gated networks of [69], [51]. It was meant to extract classes from sensory data, together with some properties that characterize different instances of these classes. The key idea was to design an architecture performing dimensionality reduction by learning along meaningful manifolds, but representing these manifolds in such a way that it would result in a simple metric to efficiently apply unsupervised learning methods. The central assumption was that the position of an instance along the sub-manifold corresponding to its class could appropriately characterize its properties. This assumption provides a natural definition of categories and concepts and was shown to be efficient to reduce the complexity of algorithms in [88].

Concretely, the set of classes was learned with a softmax layer and the set of parameters with a softplus layer. The softmax layer ensured that only one class was active for a given input, whereas the softplus layer was appropriate for learning a continuous parametric space. For a given input point, the obtained class corresponded to the manifold on which this point was lying, and the obtained parameters corresponded to the location of that point on that manifold. Using unsupervised learning to learn other layers on top of the obtained representation would be more natural with this architecture because the split between the discrete set of manifolds and the parametrized position of a point over the manifold simplifies the definition of the appropriate metric. However, this further stage was not studied in [81].

A further requirement of the architecture was to endow the system with multimodal learning capabilities [81]. To ensure such capabilities, the architecture was extended as illustrated in Fig. 4 with several networks learning a representation in each modality that were then combined by sharing the softmax and softplus layers. The factored gated connections helped learning compact representations of multi-dimensional relations between different variables.

This architecture was applied to learning to classify a set of concepts, such as numbers represented as small images in the MNIST data set [95]. The modalities were the images themselves, the auditory signal corresponding to uttering the number and, more interestingly, the joint position of the iCub robot drawing the number in a kinesthetic teaching mode. The results showed that multimodal learning considerably improved the robustness of the learning process (see [81] for details).

So, to what extent does this work contribute to deep developmental robotics? In a way, it can be seen as a step towards solving the symbol grounding problem, by extracting token-like representations out of raw multimodal sensory data, while relying exclusively on unsupervised learning. However, it also suffers from severe limitations. First, the image data set (normalized images of white numbers on a black background) was designed for the specific task at hand (representing numbers), which severely contrasts with the standard developmental robotics methodology. Second, the number of concepts was predetermined and corresponded to the number of classes in the classification layer. Finally, it was unclear how to build a hierarchy of higher and higher level symbols by stacking several similar networks on top of each other. A preliminary attempt to better comply with developmental robotics constraints was published in [96], but the corresponding work does not address all requirements yet.

Thus, the representational elements described in this section may play a role in some attempts to build a deep developmental robotics architecture, but they are still preliminary and a lot of effort is required to design better technical tools in this direction.

B. Hierarchical Prediction and Temporal Clustering

There are many deep learning works dealing with temporally organized information in speech, text or videos.

However, their approach to the recognition of patterns in the corresponding flow of information is essentially the same as for static data: the network is tuned to *recognize* the pattern into a larger space where time is an additional dimension (e.g. [97], [98]). In this section, we advocate for an alternative perspective where the network is tuned to *predict* future data and we describe some works that are more in line with this perspective.

Many approaches to cognition agree on the fact that *reactive* sensorimotor information processing should be coupled with a *predictive* internal loop that has been termed in many different ways: the ideomotor principle [99], anticipatory processes [100], [101], internal simulation [102], predictive processing [103], [104], etc. As a matter of fact, many developmental psychology results show without ambiguity that infants are able to foresee the consequences of their actions before making

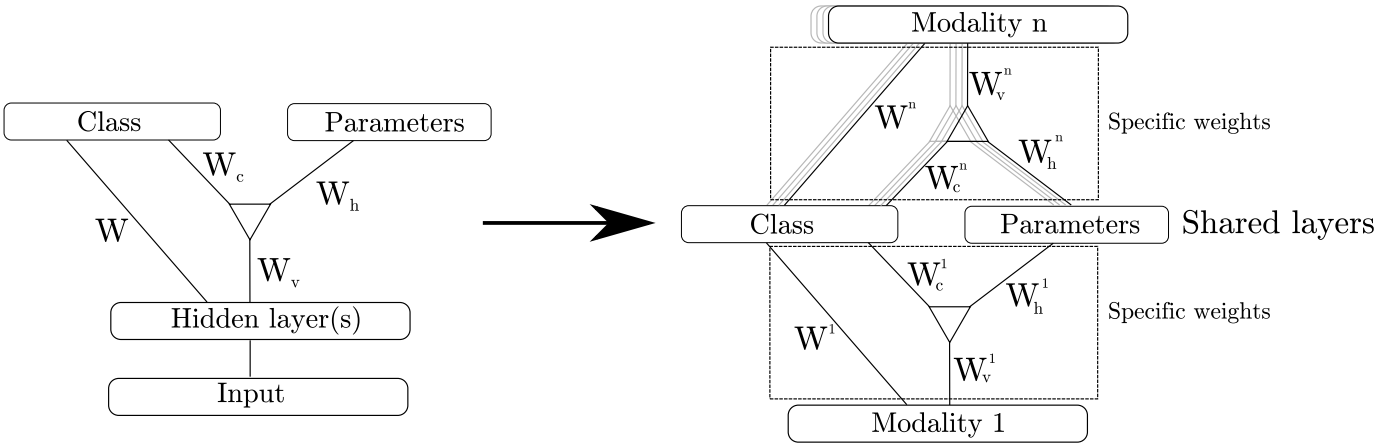

Fig. 4. The architecture depicted in Fig. 3 generalizes to an arbitrary number of modalities by duplicating the input layers while sharing the same softmax and softplus layers (*Modality 1* has been separated for clarity only, all modalities are equivalent in the network). This constrains the network to learn a joint representation of different modalities, which have to be clustered into a single space (one softmax unit active at one time), but the network can allocate different softplus units to one or several modalities, according to their correlations and the number of available units.

a choice, and such anticipatory capabilities are considered central to intelligence in many theories, e.g. [105], [106], [107].

A consequence of the predictive framework for pattern recognition is that the incoming stimuli are interpreted in the context of internally generated predictions [108]. With this perspective, ”a perceptual process does not start with the stimulus; rather, the stimulus is an END of the process, like the last piece of a jig-saw puzzle, which fits in its place only because all the other pieces have been placed in a particular way. [...] A stimulus is present only if there is an organization into which it can be fitted” [109].

To support internal simulation, it is necessary to learn representations of what will happen next given the current context, eventually including some memory from the past. But it is most probably not the case that we mentally simulate all the instants from the current situation to the desired state. We certainly make profit of a hierarchically organized temporal structure to skip many unnecessary details in this mental simulation process [102]. So, on top of the lowest level of basic entities such as sensorimotor contingencies or the sensory effect of actions on objects, the knowledge of a robot should be endowed with a hierarchical structure not only over its representational content, but also over time. This *hierarchical prediction* issue has received a lot of attention recently, see e.g. [110], [104] and the corresponding capabilities have been named “hierarchical predictive processing” [103].

A second issue is that, even at the lowest level of the hierarchy, the elementary entities are not static patterns, they have a temporal extension. For instance, when you are pushing an object, it may roll away from your finger, it may move in contact with it, it may fall aside or it may even not move at all if it is fixed or too heavy [22]. Figuring out these effects requires observing over time. Furthermore, the action of pushing itself must last long enough for perceiving these effects. Recognizing these various effects requires clustering over some undetermined interval of time, since these effects and the corresponding actions are not instantaneous. We call this the *temporal clustering* issue.

Both issues, hierarchical prediction and temporal clustering, are key challenges for developmental robotics. So, given the focus of this paper on the contribution of deep learning techniques to developmental robotics, we refer to deep learning techniques that we believe should contribute to solving these issues.

To approach the temporal clustering issue, one can consider that, when what happens immediately next is predictable, there some internal coherence into the temporal data, so it pertains to the same temporal unit [111], [112]. Additionally, in order to predict the next pattern from the current situation, instead of learning a large set of associations between successive patterns, one should learn the *transformations* from given patterns to their successors. So, one may solve the temporal clustering issue by learning transformations from a sensation to the next and considering that the flow of sensorimotor information pertains to the same temporal unit as long as the transformation is the same.

A first deep learning attempt to learn such transformations between images was based on the network illustrated in Fig. 2 [51]. A feature sharing property was introduced, as explained in Section II-D, leading to a degenerate case of the network first published in [69]. In order to learn transformations with this network, the “x” layer is associated to the current pattern, the “y” layer to the next pattern and the “z” layer to the transformation between the two. The transformations themselves are coded as a finite repertoire where each neuron in the “z” layer conveys a binary latent variable that corresponds to a components of the transformation.

A further improvement to this approach consisted in customizing the structure and learning mechanisms of the architecture so that it would preferentially extract *orthogonal* transformations [113]. An important property of orthogonal transformations is that there is no information loss between the initial and the final data. This is the case for instance for rotation of images. So, a loss of orthogonality in a sequence of input data can be interpreted as a meaningful change in

the content represented by the data. Therefore, detecting this change might be used to perform temporal clustering.

The second ingredient to perform hierarchical prediction consists in storing temporally extended entities through some kind of internal memory. One line of research in the deep learning literature in this domain results from the augmentation of the RBM architecture to deal with time. A first approach was the CRBM architecture, which consisted of RBMs with additional input from the previous time frame [114]. The gated, transformation-based architectures described in Section II-D are indirect descendants of this approach through a series of evolutions [115], [116]. Another approach was the TRBM architecture [97] which was composed of a succession of RBMs for successive time frames, the bias on hidden units of the next RBM being a function of the hidden activation at the previous time frame. However, this model was suffering from an inefficient inference process and was later improved with the RTRBM, where the dependency between two time frames can be represented in practice with a recurrent connection [63]. For more recent work along this line, see [117].

However, even if these approaches could in principle model regularities over many time steps, in practice they were only successful over a limited time window, due to the *vanishing or exploding gradient problem* [33]. In the context of recurrent neural networks (), this problem was specifically preventing from learning long term dependencies in a sequence of input.

The Long-Short Term Memory (LSTM) approach was designed specifically to remove this limitation [118]. The idea of the LSTM architecture was to include into the network some units dedicated to maintaining an appropriate amount of error back-propagation. Then, those units had to be isolated from the standard gradient back-propagation flow with an "input gate" and an "output gate" whose opening and closing dynamics was learned. This made the network able to learn long term sequential problems much better than most previous approaches. A "forget gate" was then added to deal with the necessity of reinitializing the content of the internal state when the context was changing [119]. This successful architecture is the starting point of a large body of work where sequences are learned. Among other things, the initially purely forward learning mechanism was replaced by a bidirectional learning mechanism to improve performance in contexts where the overall sequence is stored and accessible in advance, such as in text or speech processing [120], [121]. Furthermore, hierarchical stacking of several LSTMs improved the capability to learn complicated sequences [122].

Another, fast growing line of research has been initiated with "Multiplicative " [123], which combine with the kind of gating architecture described in Section II-D, and benefit from Hessian-free optimization [46].

In the same domain, an even more recent line of research emerged with two different architectures. One is based on a set of "clocks" constraining the dynamics of recurrent units at different frequencies [124], with competitive performance with respect to the works presented above. The second implements a similar idea using different decay rates in the activity of recurrent units [125].

To conclude this section, it seems that some deep learning techniques offer a few of the required properties to address hierarchical prediction and temporal clustering, but the concrete way to combine the ingredients described above to actually build hierarchies of temporal representations is still the focus of very intensive research [126], [98], [127].

V. Behavior optimization, intrinsic motivations and life-long learning

All the work we have surveyed so far was dedicated to endowing developmental robotics architectures with the elaborated sub-symbolic representations they are missing. But learning representations is not enough. A developmental robot must act in its environment and improve its behavior along time. By improving, we mean that, with more experience, it should minimize some implicit or explicit cost function.

In this section, we investigate a family of mechanisms that should play a role in experience-based improvement, still from the dual perspective of the impact of deep learning techniques on those mechanisms and of the impact of those concerns on the future of deep learning research.

A. Optimizing control and actions

Like robotics perception, robotics behavior is subject to the curse of dimensionality. Indeed, complex robots such as humanoids have many degrees of freedom. Learning to act or behave can be cast as an optimization problem that is generally very expensive in the size of the search space. So, trying to learn the appropriate behaviors by directly searching the huge space of all possible movements is deemed to fail. Given this complexity, it has been recognized that the data corresponding to useful actions lies in some limited sub-manifolds of the motor space [128]. Indeed, an autonomous robot should often reproduce the same patterns of movements in many everyday-life contexts and this limited set of movements should only cover a small portion of the space it can actually reach.

In the standard robot control learning literature, this recognition has favored the emergence of an approach based on the optimization of *movement primitives*, i.e. mathematical abstractions that define specific movements in small spaces. A popular example of such primitives are Dynamic Movement Primitives (DMPs) [129], [130].

A DMP generates a movement from a starting point to a goal. In order to circumvent the curse of dimensionality, it is generally parametrized only with time. As such, it can be seen as a way to realize a unique instance of a motor primitive that works only in very specific circumstances [131]. In order to increase its expressive power, it can be augmented into a *contextual DMP*. Contextual DMPs are DMPs whose representation has been extended with additional parameters such as the goal, the starting point, some waypoints or any contextual constraints that can be exerted on a movement [132]. Learning a contextual DMP means learning its representation for different values of the contextual parameters and then relying on the generalization capability of some regression algorithm [133] to provide a correct behavior, even for contexts in which the system was not trained. There are several representations for contextual DMPs [134], [135],

[131], but their common feature is that the dimensionality grows quickly with the number of contexts that have to be distinguished. As a consequence, optimizing contextual DMPs after learning them from demonstrations faces the curse of dimensionality again. And the parameters of a contextual DMP have to be determined by the designer, which contradicts the requirements of developmental robotics.

As we outlined in Section IV-A, some deep learning techniques are readily endowed with the capability to focus on the relevant sub-manifolds of a large domain. Thus they may provide the adequate tools for learning to cover the parts of the sensorimotor domain corresponding to useful postures or behaviors without the need for the engineer-based pre-structuring provided by DMPs.

Along this line, deep learning techniques were used in a simulated race cars driving experiment to compactly encode the state space of a control problem using sparse auto-encoders [136], [137]. A more recent example in the same domain used CNNs (see Section II-A) for the same purpose [138].

Apart from structuring a continuous state-action domain, deep learning techniques can also be used to learn an accurate model of a performance function, the *value function*, that is at the heart of *Reinforcement Learning* (RL) algorithms [139]. Reinforcement Learning addresses the problem of improving the behavior of a system based on some external cost or reward function. For a given state, the *value function* tells you how good it is to be in that state, so it helps you choosing the next state you should reach. When the action domain is continuous, as is the case when controlling robots, choosing the best action becomes a costly optimization problem. In principle, the value function helps reducing this cost, but the standard linear methods suffer from the difficulty to approximate it with linear approximation tools over a continuous domain [140]. As a consequence, RL for robotics control evolved towards pure black-box, stochastic optimization methods, that do not strive to approximate the value function [141], [131].

So far, most attempts to combine RL with deep neural networks have been applied in domains that have few to do with developmental robotics. A survey of those attempts is provided in [142]. But the situation is evolving quickly. Actually, deep learning techniques can approximate the value function of an RL algorithm much more accurately than standard method because they find the adequate features for doing so. This insight led to a first breakthrough when a CNN was combined with a slightly modified version of the standard Q-LEARNING algorithm [139] to get impressive results in a large set of challenging video game problems [143]. Then, very recently, the method was applied to twenty continuous action problems in an actor-critic architecture [144] and should result in the revival of attempts to approximate the value function of RL algorithms into developmental robotics problems.

Another line of research consists in applying *guided policy search* techniques [145] to learn a high-level, decision policy from raw image data, giving rise to "end-to-end" learning of visuomotor policies using CNNs [146]. Intuitively, the capability of deep learning techniques to build hierarchical representations plays a crucial role in such attempts to integrate control and decision into a unified framework. A new line of research now consists in combining this policy improvement issue with the issue of dealing with sequential data as outlined in Section IV-B. Such combinations are important steps towards the emergence of cognitive operators, as we outlined in the introduction of this section.

B. From intrinsic motivations to curriculum learning

Like infants during their development, developmental robots should face a complex, continuously changing environment, with a huge space of opportunities for perceiving, acting and learning. In such a space, choosing what to learn next is itself a hard problem that should not be solved through *ad hoc* engineering methods, but rather in a principled way. Such a principled approach relies on intrinsic motivations and artificial curiosity (see [147] for a survey).

Intrinsic motivations stand for the fact that, rather than learning what to do next based directly on some external or extrinsic reward signal, we learn based on some internal or intrinsic motivational system that is only indirectly related to the external reward signal. One specific intrinsic motivation is curiosity. The curiosity motivation stands for the tendency of infants to spend a lot of time exploring the outcome of their activities in a way that may seem random at first glance [78], but that is in fact organized so as to maximize their global learning speed.

When transposed to robotics or software agents, all the Intelligent Artificial Curiosity (IAC) architectures share a common principle: given a discrete repertoire of potential actions, the IAC process selects the action that maximizes the progress measurement of the robot or agent, i.e. the action for which an approximation of the performance derivative is maximal.

Several computational models have been proposed to implement IAC, e.g. [148], [149], [150]. In all these models, the perceptive space of the agent has to be clustered into domains where the performance progress is measured. In [151], this clustering is performed hierarchically. Thus the learning progress measurement processes of IAC may benefit from the outstanding hierarchical clustering capabilities of deep learning techniques. Reciprocally, deep learning techniques may benefit for the increased learning speed of IAC[1]. To our knowledge, such a combination has not been attempted so far.

In direct connection with this issue, Piaget showed how the development of infants is structured into stages, though the exact order of the learning stages is specific to each infant [152]. In particular, infants seem to first learn easy things and progressively turn towards harder activities as they acquire more complex representations and capacities. Accounting for this *developmental trajectory* and explaining the increasing complexity of the problems an agent can address through its lifetime is the central concern of the so-called "life-long learning" research domain. Furthermore, the dynamics

[1]Actually, using typical developmental processes such as IAC to improve deep learning techniques should give rise to *developmental deep learning* rather than *deep developmental learning*, but we do not cover this potential line of research any further here.

of this complexity growth should be partly conditioned by maturational constraints related to the biological growth of the agent, as put forward in [153], [151].

In the deep learning literature, these issues have started being investigated under the label "curriculum learning" [154], but a lot remains to be studied (see Section 9.1.4 of [24] for an overview).

VI. Conclusion and perspectives

Starting from the fact that a lot of engineering is still required to extract features for the representation of affordances, we outlined that it would be straightforward to use deep learning techniques for the perception of robots and use some other techniques for behavioral learning based on the obtained perceptive representations. However, such an approach would maintain a questionable split between the perceptive, decisional and behavioral capabilities of the robot. Despite numerous criticisms against this deeply rooted split [155], [72], [156], a satisfactory alternative is still missing.

In this paper, through a survey of deep learning techniques that may be used in developmental robotics, we have tried to investigate some foundations towards such a satisfactory alternative. We have investigated the potential impact of deep learning techniques on building representations for robots based on pattern recognition methods. We have also examined diverse issues related to behavior optimization, as well as natural language and other socially oriented learning processes. From a more methodological standpoint, finally, we have outlined the necessity to rely on unsupervised learning and hierarchical predictive processes.

Some of the elements described in this paper may be used to build better foundations for developmental robotics. However, many issues remain to be addressed. As a final word, we focus on two of them: the current technical difficulty of running deep learning algorithms onto robots, and the need for a general architecture combining all the building blocks we studied separately so far.

A. Towards deep learning embedded in robots

Though survey paper dedicated to deep learning techniques generally mention robotics as a potential application domain, so far there are surprisingly few papers where deep learning techniques are actually used in a robotics context, despite a few prominent and recent exceptions, e.g. [146], [157], [28].

One reason for this situation is that, in practice, deep architectures are usually computationally expensive to train and tune. Actually, to get some of the most impressive non-robotics pattern recognition results, very large clusters of computers were used (e.g. [158]). Thus, embedding enough computational power on a robot to process on-line all raw sensory information with the deep learning techniques described throughout this paper is probably still out of reach [159]. However, from the purely computational point of view, the situation should improve quickly because deep networks are highly parallelizable and make good use of computing resources, and better and better GPU resources become available.

Nevertheless, in most robotics experiments, data is processed offline, which may seem incompatible with the incremental and on-line learning requirements of developmental robotics. As outlined in Section II-B, deep neural networks are generally trained layer per layer, often reusing the same input data for training the added layers in sequence. One may argue that this training process contradicts the requirements of developmental robotics because, according to the life-long learning perspective, the learner should be trained by the incoming data arising from its own experience rather than through successive training steps managed by the experimenter.

A complementary perspective on that issue is that processing everything on-line is not mandatory. In humans and animals, some learning takes place during sleep [160] based on some *experience replay* mechanism [161], [162] and may involve some internal reorganization of the input flow. Thus, it is not so unrealistic to unsupervised layer-per-layer training of a deep network so as to implement "representational redescription" of sensorimotor data. This redescription of data acquired during "day-time" may happen during "night-time", when the robot is available for batch processing. This does not eliminate the need for purely immediate, on-line learning during the activities of the robot, but this provides a solution to implement the most expensive representation learning processes in a batch or mini-batch way. This approach is under current investigation in the DREAM project [163] which supports some work described in this paper.

B. Step towards a general architecture

How can we put together all the building blocks we have described separately so far?

First, the architecture should extract from the sensorimotor flow of a robot a set of sensorimotor contingencies, as outlined in Section III-A, that would correspond to transformations of raw sensations in this flow, building on techniques described in Section IV-B. Taking these sensorimotor contingencies as input, it should then implement hierarchical predictive processes. We outlined in Section IV-B that a lot of research is still necessary to endow a robotic system with the corresponding capabilities. This approach might be considered successful if the robot could prove capable of building sub-symbolic representations for affordances, objects, actions and effects into a hierarchy corresponding to the kind of sub-symbolic knowledge we have outlined in the introduction. Note that, at this stage, the role in this perspective of the conceptual clustering processes over manifolds, described in Section IV-A, is not clear yet and also deserves significantly more research.

A second and even more challenging stage would then consist in integrating the behavior optimization and curiosity mechanisms described in Section V. These mechanisms are required to improve the realization of actions, the choice of an action or the activation of an affordance in a given context. But they can also contribute to "sculpting" the sensorimotor and sub-symbolic representations by orienting the selection of features towards those that are useful for efficient behavior. Here, the unsupervised learning processes that are responsible

for extracting a hierarchy of features should be reconsidered so that the features they choose are those that best support the improvement of behavior.

But, if we come back to the mug example presented in the introduction, the sensorimotor and sub-symbolic representations we would obtain would account for part of the example, such as the knowledge of how it feels to lift it or the proper way to handle it, but it may not be enough to account for more symbolic aspects such as naming the object or eventually knowing where to purchase one.

Building symbolic knowledge on top of the sensorimotor and sub-symbolic representations as well as learning cognitive operators to manipulate temporal information are still open research questions. Natural language acquisition and human-robot interaction processes should definitely come into play, because cultural, external representations are most probably necessary to reach this stage [164], and language and interaction are the appropriate media to learn about these external representations. Learning such representation might be supported by dedicated processes, given the specific educational effort it takes to a child to learn them. But an architecture including them might not be strictly hierarchical, given the impact of top-down processes in the organization of the hierarchy of features and operators. In particular, these processes may contribute to endowing a developmental robotics architecture with reasoning and planning capabilities beyond what hierarchical prediction processes may offer. They may also help to build more abstract cognitive capabilities on top of embodied cognition, shifting from the representations of concepts that clearly have some modal correlates to more abstract notions such as "liberty" or "institution".

So, beyond the mere integration of various deep learning "modules" dedicated to specific capacities such as image processing, multimodal fusion, natural language understanding or social signal processing, what is required is a deeper integration of basic machine learning capabilities into a set of well chosen deep learning tools, so that all the above capacities emerge from the sub-symbolic representations learned with these tools. This is this ambitious research program that we called deep developmental learning, and that should result in the definition of a deep developmental robotics architecture.

Acknowledgments

This work was partially supported by the European Commission, within the CoDyCo project (FP7-ICT-2011-9, No.600716), and by the European Union's Horizon 2020 research and innovation program within the DREAM project under grant agreement No 640891. We thank two anonymous reviewers for their helpful comments, as well as Stéphane Doncieux, Ryan Lober, Antonin Bertin and Pierre-Yves Oudeyer for useful discussions about the content of this article.

References

[1] A. Newell, "Physical symbol systems," *Cognitive Science*, vol. 4, pp. 135–183, 1980.

[2] J. R. Searle, "Minds, brains, and programs," *Behavioral and Brain Sciences*, vol. 3, no. 03, pp. 417–424, 1980.

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

[4] A. Clark, *Being there: Putting brain, body, and world together again*. MIT press, 1997.

[5] ——, *Supersizing the mind: Embodiment, action, and cognitive extension: Embodiment, action, and cognitive extension*. Oxford University Press, 2008.

[6] S. Lemaignan, R. Ros, L. Mosenlechner, R. Alami, and M. Beetz, "Oro, a knowledge management platform for cognitive architectures in robotics," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3548–3553.

[7] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, no. 2, pp. 185–193, 2001.

[8] 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.

[9] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.

[10] J. Piaget, *The origins of intelligence in children.* WW Norton & Co, 1952.

[11] L. S. Vygotsky, "Play and its role in the mental development of the child," *Journal of Russian and East European Psychology*, vol. 5, no. 3, pp. 6–18, 1967.

[12] P. H. Miller, *Theories of developmental psychology*, 5th ed. Worth Publishers, 2010.

[13] A. N. Meltzoff, P. K. Kuhl, J. Movellan, and T. J. Sejnowski, "Foundations for a new science of learning," *Science*, vol. 325, no. 5938, pp. 284–288, 2009.

[14] A. Stoytchev, "Some basic principles of developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 122–130, 2009.

[15] F. Guerin, N. Kruger, and D. Kraft, "A survey of the ontogeny of tool use: from sensorimotor experience to planning," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 1, pp. 18–45, 2013.

[16] J. J. Gibson, *The Theory of Affordances*, R. Shaw and J. Bransford, Eds. Hillsdale, New jersey: John Wiley and Sons, 1977.

[17] E. Ugur and J. Piater, "Emergent structuring of interdependent affordance learning tasks," in *International Conference on Development and Learning and on Epigenetic Robotics*, 2014.

[18] E. E. Aksoy, M. Tamosiunaite, R. Vuga, A. Ude, C. Geib, M. Steedman, and F. Wörgötter, "Structural bootstrapping at the sensorimotor level for the fast acquisition of action knowledge for cognitive robots," in *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics*. IEEE, 2013, pp. 1–8.

[19] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 3060–3065.

[20] E. Ugur, E. Sahin, and E. Oztop, "Affordance learning from range data for multi-step planning," in *International Conference on Epigenetic Robotics*, 2009.

[21] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4373–4378.

[22] E. Ugur, S. Szedmak, and J. Piater, "Bootstrapping paired-object affordance learning with learned single-affordance features," in *International Conference on Development and Learning and on Epigenetic Robotics*, 2014.

[23] G. Pezzulo, L. W. Barsalou, A. Cangelosi, M. H. Fischer, K. McRae, and M. J. Spivey, "The mechanics of embodiment: A dialog on embodiment and computational modeling," *Embodied and grounded cognition*, p. 196, 2011.

[24] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[25] Y. LeCun, J. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[26] M. Zorzi, A. Testolin, and I. P. Stoianov, "Modeling language and cognition with deep unsupervised learning: a tutorial overview," *Frontiers in psychology*, vol. 4, 2013.

[27] I. P. Stoianov and M. Zorzi, "Emergence of a 'visual number sense' in hierarchical generative models," *Nature neuroscience*, vol. 15, no. 2, pp. 194–196, 2012.

[28] A. Di Nuovo, M. Vivian, and A. Cangelosi, "A deep learning neural network for number cognition: a bi-cultural study with the icub," in *ICDL-EPIROB*, 2015.

[29] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[30] O. Delalleau and Y. Bengio, "Shallow versus Deep Sum-Product Networks," in *Advances in Neural Information Processing Systems (NIPS) 24*, 2011, pp. 666–674.

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

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A field guide to dynamical recurrent neural networks*. IEEE Press, 2001.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, 2010, pp. 249–256.

[35] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[36] A. Salman and K. Chen, "Exploring speaker-specific characteristics with deep learning," in *The 2011 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2011, pp. 103–110.

[37] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[38] A. Stuhlsatz, J. Lippel, and T. Zielke, "Discriminative feature extraction with Deep Neural Networks," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[39] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS) 25*, P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1106–1114.

[41] N. Jaitly and G. E. Hinton, "Learning a better representation of speech soundwaves using restricted boltzmann machines," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5884–5887.

[42] A.-R. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[43] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 5, 2009.

[44] P. Mehta and D. J. Schwab, "An exact mapping between the variational renormalization group and deep learning," *arXiv preprint arXiv:1410.3831*, 2014.

[45] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," *arXiv preprint arXiv:1503.02406*, 2015.

[46] J. Martens, "Deep learning via hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 735–742.

[47] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep Big Multilayer Perceptrons For Digit Recognition," *Neural Networks Tricks of the Trade*, vol. 1, pp. 581–598, 2012.

[48] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-Column Deep Neural Network for Traffic Sign Classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[49] G. Alain and Y. Bengio, "What Regularized Auto-Encoders Learn from the Data Generating Distribution," *arXiv*, vol. 1211.4246v4, 2013.

[50] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, 2010.

[51] R. Memisevic, C. Zach, G. Hinton, and M. Pollefeys, "Gated softmax classification," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., 2010, pp. 1603–1611.

[52] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *Computational Intelligence Magazine, IEEE*, vol. 5, no. 4, pp. 13–18, 2010.

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

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

[55] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[57] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 833–840.

[58] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot, "Higher Order Contractive Auto-Encoder," in *ECML/PKDD (2)*, 2011, pp. 645–660.

[59] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, vol. 19, 2007, p. 801.

[60] P. Smolensky, "Information processing in dynamical systems: foundations of harmony theory," in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*. MIT Press, 1986, pp. 194–281.

[61] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[62] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[63] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The Recurrent Temporal Restricted Boltzmann Machine." in *Advances in Neural Information Processing Systems (NIPS) 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2008, pp. 1601–1608.

[64] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[65] M. Ranzato, Y.-l. Boureau, S. Chopra, and Y. Lecun, "A unified energy-based framework for unsupervised learning," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 371–379.

[66] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Advances in Neural Information Processing Systems*, 2013, pp. 899–907.

[67] J. Rudy and G. W. Taylor, "Generative class-conditional autoencoders," *arXiv preprint arXiv:1412.7009*, 2014.

[68] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 5, 2009, pp. 448–455.

[69] R. Memisevic and G. E. Hinton, "Learning to represent spatial transformations with factored higher-order boltzmann machines," *Neural Computation*, vol. 22, no. 6, pp. 1473–1492, 2010.

[70] R. Memisevic, "Gradient-based learning of higher-order image features," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 1591–1598.

[71] ——, "Learning to relate images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1829–1846, 2013.

[72] J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness," *Behavioral and Brain Sciences*, vol. 24, no. 05, pp. 939–973, 2001.

[73] A. Laflaquiere and N. Hemion, "Grounding object perception in a naive agents sensorimotor experience," in *ICDL-EPIROB*, 2015.

[74] D. Jayaraman and K. Grauman, "Learning image representations equivariant to ego-motion," *arXiv preprint arXiv:1505.02206*, 2015.

[75] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[76] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[77] E. Koechlin, C. Ody, and F. Kouneiher, "The architecture of cognitive control in the human prefrontal cortex," *Science*, vol. 302, no. 5648, pp. 1181–1185, 2003.

[78] L. Smith and M. Gasser, "The development of embodied cognition: Six lessons from babies," *Artificial life*, vol. 11, no. 1-2, pp. 13–29, 2005.

[79] H. McGurck and J. W. McDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 246-248, 1976.

[80] M. Botvinick and J. Cohen, "Rubber hands 'feel' touch that eyes see." *Nature*, vol. 391, no. 6669, p. 756, Feb. 1998.

[81] A. Droniou, S. Ivaldi, and O. Sigaud, "A deep unsupervised network for multimodal perception, representation and classification," *Robotics and Autonomous Systems*, p. to appear, 2014.

[82] S. Ivaldi, S. M. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, P.-Y. Oudeyer, and O. Sigaud, "Object learning through active exploration," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 1, pp. 56–72, 2014.

[83] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, *et al.*, "Recent advances in deep learning for speech research at microsoft," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8604–8608.

[84] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5532–5536.

[85] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante, "Chalearn gesture challenge: Design and first results," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2012, pp. 1–6.

[86] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 474–490.

[87] L. Cayton, "Algorithms for manifold learning," University of California, San Diego, Tech. Rep., 2005.

[88] H. Narayanan and S. Mitter, "Sample complexity of testing the manifold hypothesis," in *Advances in Neural Information Processing Systems (NIPS) 23*, 2010.

[89] S. Rifai, Y. N. Dauphin, P. Vincent, Y. Bengio, and X. Muller, "The Manifold Tangent Classifier," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 2294–2302.

[90] S. Reed and H. Lee, "Learning Deep Representations via Multiplicative Interactions between Factors of Variation," in *Workshop NIPS*, 2013.

[91] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[92] J. Macqueen, "Some methods for classification and analysis," in *Berkeley Symposium on Mathematical Statistics and Probability*, vol. 233, 1967, pp. 281–297.

[93] J. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[94] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 69, pp. 59–69, 1982.

[95] Y. LeCun and C. Cortes, "The mnist database of handwritten digits," 1998.

[96] A. Droniou, S. Ivaldi, and O. Sigaud, "Learning a repertoire of actions with deep neural networks," in *ICDL-Epirob*, 2014, pp. 1–6.

[97] I. Sutskever and G. E. Hinton, "Learning Multilevel Distributed Representations for High-dimensional Sequences," *Proceeding of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[98] V. Michalski, R. Memisevic, and K. Konda, "Modeling deep temporal dependencies with recurrent "grammar cells"," in *Advances in neural information processing systems*, 2014, pp. 1925–1933.

[99] G. Pezzulo, G. Baldassarre, M. V. Butz, C. Castelfranchi, and J. Hoffmann, "From actions to goals and vice-versa: Theoretical analysis and models of the ideomotor principle and tote," in *Anticipatory Behavior in Adaptive Learning Systems*. Springer, 2007, pp. 73–93.

[100] M. V. Butz, O. Sigaud, and P. Gérard, "Anticipatory behavior: Exploiting knowledge about the future to improve current behavior," in *Anticipatory behavior in adaptive learning systems*. Springer, 2003, pp. 1–10.

[101] ——, "Internal models and anticipations in adaptive learning systems," in *Anticipatory behavior in adaptive learning systems*. Springer, 2003, pp. 86–109.

[102] G. Pezzulo and C. Castelfranchi, "Thinking as the control of imagination: a conceptual framework for goal-directed systems," *Psychological Research*, vol. 73, no. 4, pp. 559–577, 2009.

[103] A. Clark, "Whatever next? predictive brains, situated agents, and the future of cognitive science," *Behavioral and Brain Sciences*, vol. 36, no. 03, pp. 181–204, 2013.

[104] K. Friston and S. Kiebel, "Predictive coding under the free-energy principle," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1521, pp. 1211–1221, 2009.

[105] A. Berthoz, *Le sens du mouvement*. Paris: Éditions Odile Jacob, 1997.

[106] J. Hawkins and S. Blakeslee, *On intelligence*. Macmillan, 2007.

[107] W. Gaona, E. Escobar, J. Hermosillo, and B. Lara, "Anticipation by multimodal association through an artificial mental imagery process," *Connection Science*, vol. 27, no. 1, pp. 1–21, 2014.

[108] G. Pezzulo, "Coordinating with the future: the anticipatory nature of representation," *Minds and Machines*, vol. 18, no. 2, pp. 179–225, 2008.

[109] T. Jarvilehto, "Efferent influences on receptors in knowledge formation," *Psycoloquy*, vol. 9, p. 41, 1998.

[110] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[111] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural Networks*, vol. 16, no. 1, pp. 11–23, 2003.

[112] J. A. Starzyk and H. He, "Anticipation-based temporal sequences learning in hierarchical structure," *Neural Networks, IEEE Transactions on*, vol. 18, no. 2, pp. 344–358, 2007.

[113] A. Droniou and O. Sigaud, "Gated autoencoders with tied input weights," in *International Conference on Machine Learning*, 2013, pp. 1–6.

[114] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Advances in neural information processing systems*, 2006, pp. 1345–1352.

[115] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 1025–1032.

[116] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series," *The Journal of Machine Learning Research*, vol. 12, pp. 1025–1068, 2011.

[117] L. Pasa, A. Testolin, and A. Sperduti, "Neural networks for sequential data: a pre-training approach based on hidden markov models," *Neurocomputing*, 2015.

[118] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in *Advances in Neural Information Processing Systems 9: Proceedings of the 1996 Conference*, vol. 9. MIT Press, 1997, p. 473.

[119] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[120] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*. Springer, 2005, pp. 799–804.

[121] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[122] S. Fernández, A. Graves, and J. Schmidhuber, "Sequence labelling in structured domains with hierarchical recurrent neural networks." in *IJCAI*, 2007, pp. 774–779.

[123] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1017–1024.

[124] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork rnn," *arXiv preprint arXiv:1402.3511*, 2014.

[125] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M. Ranzato, "Learning longer memory in recurrent neural networks," *arXiv preprint arXiv:1412.7753*, 2014.

[126] K. R. Konda, R. Memisevic, and V. Michalski, "Learning to encode motion using spatio-temporal synchrony," *arXiv preprint arXiv:1306.3162*, 2013.

[127] D. C. Mocanu, H. B. Ammar, D. Lowet, K. Driessens, A. Liotta, G. Weiss, and K. Tuyls, "Factored four way conditional restricted boltzmann machines for activity recognition," *Pattern Recognition Letters*, 2015.

[128] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle synergies in the construction of a natural motor behavior," *Nature neuroscience*, vol. 6, no. 3, pp. 300–308, 2003.

[129] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[130] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[131] F. Stulp and O. Sigaud, "Robot skill learning: From reinforcement learning to evolution strategies," *Paladyn Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, august 2013.

[132] F. Stulp, G. Raiola, A. Hoarau, S. Ivaldi, and O. Sigaud, "Learning compact parameterized skills with expanded function approximators," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 1–7.

[133] F. Stulp and O. Sigaud, "Many regression algorithms, one unified model: A review," *Neural Networks*, vol. 69, pp. 60–79, 2015.

[134] B. Nemec, M. Tamosiunaite, F. Wörgötter, and A. Ude, "Task adaptation through exploration and action sequencing," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009.

[135] J. Kober, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[136] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[137] ——, "Deep learning of visual control policies." in *ESANN*, 2010.

[138] J. Koutník, J. Schmidhuber, and F. Gomez, "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning," in *Proceedings of the 2014 conference on Genetic and evolutionary computation*. ACM, 2014, pp. 541–548.

[139] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[140] M. Lagoudakis and R. Parr, "Least square policy iteration," *Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.

[141] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012, pp. 1–8.

[142] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[143] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[144] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[145] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, 2014, pp. 1071–1079.

[146] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *arXiv preprint arXiv:1504.00702*, 2015.

[147] P.-Y. Oudeyer, A. Baranès, and F. Kaplan, "Intrinsically motivated learning of real-world sensorimotor skills with developmental constraints," in *Intrinsically motivated learning in natural and artificial systems*. Springer, 2013, pp. 303–365.

[148] J. Schmidhuber, "Curious model-building control systems," in *IEEE International Joint Conference on Neural Networks*. IEEE, 1991, pp. 1458–1463.

[149] P.-Y. Oudeyer, F. Kaplan, and V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[150] M. Lopes and P. Oudeyer, "Active learning and intrinsically motivated exploration in robots: Advances and challenges (guest editorial)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 65–69, 2010.

[151] A. Baranès and P.-Y. Oudeyer, "R-iac: Robust intrinsically motivated exploration and active learning," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 3, pp. 155–169, 2009.

[152] J. Piaget, *The construction of reality in the child*. Routledge, 1954, vol. 82.

[153] A. Baranès and P.-Y. Oudeyer, "The interaction of maturational constraints and intrinsic motivations in active motor development," in *IEEE International Conference on Development and Learning (ICDL)*, vol. 2. IEEE, 2011, pp. 1–8.

[154] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[155] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, pp. 14–23, 1986.

[156] O. L. Georgeon, J. B. Marshall, and R. Manzotti, "ECA: An enactivist cognitive architecture based on sensorimotor modeling," *Biologically Inspired Cognitive Architectures*, vol. 6, pp. 46–57, 2013.

[157] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[158] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8595–8598.

[159] Y. Bengio, "Deep learning of representations: Looking forward," in *Statistical language and speech processing*. Springer, 2013, pp. 1–37.

[160] U. Wagner, S. Gais, H. Haider, R. Verleger, and J. Born, "Sleep inspires insight," *Nature*, vol. 427, no. 6972, pp. 352–355, 2004.

[161] W. E. Skaggs and B. L. McNaughton, "Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience," *Science*, vol. 271, no. 5257, pp. 1870–1873, 1996.

[162] A. Peyrache, M. Khamassi, K. Benchenane, S. I. Wiener, and F. P. Battaglia, "Replay of rule-learning related neural patterns in the prefrontal cortex during sleep," *Nature neuroscience*, vol. 12, no. 7, pp. 919–926, 2009.

[163] "The dream h2020 project," http://www.robotsthatdream.eu/.

[164] D. E. Rumelhart, P. Smolensky, J. L. McClelland, and G. E. Hinton, "Sequential thought processes in PDP models," in *Parallel Distributed Processing*, 1986, ch. 14, pp. 7–57.

Alain Droniou Alain Droniou graduated from the Ecole Polytechnique, France, in 2011 and received his Ph.D. degree in computer science from the Universit Pierre et Marie Curie, France, in 2015. His research interests included humanoid robotics, developmental learning and deep neural networks. He is currently working for the French Defence procurement agency.

Olivier Sigaud Olivier Sigaud is full Professor in Computer Science at University Pierre et Marie Curie, Paris. He is centrally interested in the use of machine learning methods for developmental robotics and computational neuroscience. He was the prime investigator of the developmental robotics MACSi project (see http://http://macsi.isir.upmc.fr/) and member of the CODYCO (www.codyco.eu) and DREAM (www.robotsthatdream.eu) European projects. He also holds a PhD in philosophy since 2004. For more information, see http://people.isir.upmc.fr/sigaud.