Generating hierarchical Abstractions from sensorimotor Sequences

Mark Wernsdorfer

Bamberg University, Chair of Cognitive Systems,
Feldkirschenstr. 21, 96052 Bamberg, Germany
mark.wernsdorfer@uni-bamberg.de
http://www.uni-bamberg.de/kogsys

Abstract. One way to produce meaning within an artificial cognitive structure is to supply the system with the means necessary for generating representations autonomously. Three principally different ways of determining those means are presented and—following this distinction—a unique approach is persued. Synthesising human cognitive functions implies an understanding of how and why the human mind tends to form *hierarchical* representations of the the world. This paper develops a recursive algorithm that is capable of constructing a world model while embracing this cognitive phenomenon. Representations of concrete concepts are created from basic sensorimotor information and serve as elements for higher order representations of abstract concepts. Similar approaches of hierarchical reinforcement learning try to tackle the symbolicity of representations (e. g. context-independency or structural invariance) by categorising patterns in the temporal or spatial *structure* of sensorimotor percepts. In contrast, this paper proposes categorisations according to the *functional* aspects of memorised representations. Result is a promising framework contributing to artificial general intelligence research and artificial symbolicity/semiotics.

Keywords: emergence, hierarchical reinforcement learning, generation of symbolic representations

1 Introduction

Artificial General Intelligence (AGI) is a branch of AI research that tries to enable agents to perform domain independent intelligent action. Generalisation, however, presupposes an inductive bias in order for algorithms to pool percepts correctly according to a given domain. AGI researchers are therefore confronted with the problem of chosing a bias that allows to learn sucessfully within a maximal number of natural domains [3, pp. 303–349].

A great deal of AGI projects strives for algorithms that enable intelligent behaviour equal in or comparable to the adaptivity of humans or mammals. With respect to the fact that for generalisation, some inductive bias can never be avoided [13, pp. 20–52], the question is: How do humans generalise percepts and what is the inductive bias of human cognition?

Two main routes are treaded in order to deal with this question. The first one is by simulating a physical model of the only cognitive system that is believed to achieve general intelligence. The Blue Brain Project partially recreates the neocortex [12]—the region responsible for higher cognitive functions like anticipation and explicit memory [7, pp. 98–105]—in great detail. Over several years, extensive data from experiments on cortical tissue has been collected to generate a solid empirical ground for simulation. Because of the massive amount of data, the simulation is processed by one of IBM's BlueGene supercomputers. In case the artificial brain—with a number of neurons roughly $10^{-7}th$ of the human brain—demonstrates general intelligent behaviour at some future point in time, as a consequence from the physiological approach, inferring the human inductive bias, however, is eventually confronted with the same problems as it is concerning biological brains right now. We would have a general intelligent agent, but were unable to differentiate its sufficient causes from implementation specifities.

Another way to general intelligence is not by modeling the physical features of the mammalian neocortex but its observable functionality or abilities, like e. g. declarative and procedural memory. Those modular cognitive architectures are mostly psychologically motivated—best known representatives are Soar [10], Clarion [16] and Act-R [1]. Among those, hybrid architectures combine distributed and symbolic learning methods to automatically adapt cognitive representations to different domains. This universality is achieved by grounding representations in direct feedback from the environment (like sensor data and reward). Generated specific representations can eventually be delivered to syntactic symbol processing. Although human behavioural features are implemented by those architectures, the question remains whether they are capable of general intelligence. Furthermore, neurological findings denote that—in contrast to hybrid architectures—, the brain's functions are not spatially or temporally fixed. Instead [14] proposes that the neocortex is organised at the lowest level by one single simple algorithm. According to Mountcastle, all observable functions—in psychologically motivated architectures modeled as modules—might be emergent epiphenomena.

This paper proposes a reinforcement learning algorithm that approaches AGI by segmenting hierarchical sequences of structural and functional representations. After sketching the innovative motivation behind this approach, similar work is presented. Criteria for choosing related work are the ability to generate hierarchical representations and to segment sequences. Existing research relates foremost to hierarchical Bayesian networks and hierarchical temporal memory.

The intuition behind sequence abstraction networks is presented. The mechanism behind generating representations, resulting in a detailed model of the agent's world, is motivated. Afterwards the applied concepts are defined using common reinforcement learning terms. Pseudocode demonstrates the interaction between representations of those concepts. The novelty in the presented approach is the possibility to include functional features in the sequence recognition process.

Due to the early stage in development, instead of extensive testing and results, the last sections describe an outlook and a rough agenda for future development.

2 Hierarchical Reasoning Models with Respect for Sequenciality

This paper proposes a third route to artificial general intelligence. Neither by modeling the neural aspects of the human neocortex from third person perspective, nor by observing humans as a social counterpart from second person perspective, like in psychological empirics. Instead I describe what might be called a subjective model of the mind, that is—despite all subjectiveness—implementable, philosophically justified (epistemologically and semiotically) and already persued in a similar manner (but with a different label) by AGI researchers like Jeff Hawkins [8]. Accounted characteristica of one's own perception of the mind are *a*) a grounded and justified generation and hierarchical organisation of knowlegde, *b*) the ability to automatically and unreflectedy expect outcomes of actions and a related awareness of things "going wrong" through *c*) a memory that is queried by perceived causalities and that returns the experience of "being in a certain situation", which is not reducible to certain sensory input.[1]

2.1 Hierarchical Bayesian Networks

Bayesian Networks model combinations of dependent variables through an acyclic graph. Nodes represent random variables and edges represent conditional dependencies between those variables in problem domains. Each node stores a probability distribution of its respective variable. Parents of nodes are therefore probabilistic causes and nodes without a parent node represent independent variables.[13, pp. 184–191]

Bayesian Networks can be used for modeling inference from cause to effect or vice versa. For controlling agents in continuous domains, sequences of sensorimotor activations can be used to generate the structure of a Bayesian Network. Afterwards, desireable feedback from the environment can be triggered by executing sensorimotor activations that have been determined as causing reward.[11]

Hierarchical Bayesian Networks (HBN) extend this notion by nodes that store not only single variables but structured aggregates of variables. These variables therefore can describe Bayesian Networks themselves. The hierarchical generalisation of Bayesian Networks allows for more than propositional expression power. They are also known as Deep Belief Networks (DBN).[5]

[1] Therefore classical reinforcement learning environments, demanding for Markov descision processes [13, p. 370], are principally inappropriate to model such definition of context.

2.2 Hierarchical Temporal Memory

On the rather physiological side of modeling the brain's relevant functions is Jeff Hawkin's Hierarchical Temporal Memory (HTM). Auto-associative memory retrieval plays a crucial role in HTM, as higher order sequences return lower order sequences, that enable anticipation of environmental effects. In the Memory-Prediction Framework Hawkins sketches the cognitive capabilites of mammals as mainly an achievement of effective access to past experiences (memory)—not efficient data processing (computing).[7]

Following [14] HTM reconstructs cortical columns as elementary organisational units of the brain. In a hierarchy, this basic "neural algorithm" exhibits emergent features, like the ability for inference through invariant mental representations. Similar to HBNs HTM tries to capture temporal and spatial patterns in each node to generate high level causes unsupervisedly. As one node is succesfully able to generalise a pattern, this pattern can be referred to as an input for the generation of a higher level pattern [7]. HTMs are adapted for and extensively used in video surveillance.

3 Sequence Abstraction Network

3.1 Intuition

Main goal of the projected cognitive architecture is to generate *abstract* representations and to outline algorithmical measures for capturing *functional* similarities between representations. This contributes to the ongoing problem of generating minimally biased and invariant representations—commonly understood as being *structurally* invariant. Resulting from this implicit understanding are "vanishing intersections". Those are similar or identical percepts, that feature no structural similarity whatsoever. [6]

This paper proposes these seemingly vanishing intersections being a byproduct of restraining observations to mere structure. Including causal properties of representations—their functional role within the whole of the agent's representations—allows for comparisons along criteria like precedent or subsequent states: subjective causes and effects.

The architecture consists of a semi-supervised reinforcement learning basis that supplies the unsupervised sequence segmentation algorithm with a stream of sensorimotor activations optimised for the reward function at hand. Perceived sequences are stored in a graph-like structure that grows and fragments over time. Associated activations are passed on to a parent node as references and do themselves serve as elements of higher order sequences.

For generating these components of temporally consistent activations, two different types of descriptions are used. Ordinary similarity is calculated by referring to the *structural description* of activations—vectors in continuous sensorimotor space. *Functional descriptions* however can be collected not by breaking down the representation in structural components (single dimensions in sensorimotor space) but by considering the embedding of the representation within its neighbours sharing the same parent representation. Therefore we introduce a functional space along dimensions that define states exclusively in relation to others.

Result is a hierarchical model of the world, that generalises from concrete to abstract representations. Representations in functional space allow transfering and applying causal dependencies to structurally completely different representations in different layers of abstraction. Those different representations are related and yet they do not need to share any structural intersection. This structural and contextual independency is regarded as a first step towards artificial symbolicity/ semiotics.

3.2 Definitions

The sketched model consists of a supervised/reinforcement and a unsupervised learning algorithm. The inductive biases are sequenciality (a modification of temporal Hebbian learning [15]) and structural/functional similarity (1-nearest-neighborhood queries in Euclidean space). Those are combined to specifically reconstruct the mechanics of subjective experience.

Sensorimotor Space To avoid the transfer of semantics, the agent interfaces with the world only by perceiving continuous sensor vectors $\forall sen.sen \in \mathbb{R}^{\alpha} \wedge \alpha = dim_{\mathbb{R}}(sen)$ from sensor space S and by acting through continuous motor activations $\forall mot.mot \in \mathbb{R}^{\beta} \wedge \beta = dim_{\mathbb{R}}(mot)$ from motor space M. By integrating sensor and motor activations, the agent's interaction can be regarded as a stream of sensorimotor experience, captured by a n-dimensional vector $x \in \mathbb{R}^n$, where $n = \alpha + \beta$. The sensorimotor vector space V can therefore be described in $\mathbb{R}^n$, where $|x|$ is theoretically unlimited.

Sensorimotor vectors Sensorimotor vectors $x \in V$ within the agent's memory $V \times C$ are a concatenation of the sensor and motor activations peceived at discrete points in time. They are stored in a data structure resembling nodes in a graph. Their generation is determined by algorithm 1, which will be introduced in the next section.

Each directed edge $c \in C$ within that graph is connecting nodes holding successive vectors of sensorimotor activation. Nodes may also contain a discrete evaluation from the environment $r(W) = [-1, +1] \in \mathbb{Z}$. Sensorimotor generalisation takes place by a 1-nearest neighbour search within the generated Voronoi space [2].

World Single states w_t in the set of possible world states W can consist of all possible combinations of environment states E and agent states A ($W \equiv E \times A$). The environment is continuous, non-deterministic, only partially observable and partially controllable. In contrast, the agent's model of the world is deterministic and performs discrete transitions in state and time.

The agent receives reward from the current world state $r(w_t)$, but is only able to perceive v_t as part of the world's states $V \subset W$. Note that the observable fraction of states is not restrained to either the environment E *or* the agent A; therefore the agent may sense its own motor states as well as part of its environment.

The world changes according to $w_{t+1} = \delta(w_t)$. Changes in the world do therefore not imply action, in contrast to the classical definition in reinforcement learning ($e_{t+1} = \delta(e_t, mot_t)$, see [13, p. 370]). Because the agent has restrained sensorimotor input V, but reward $r(w_t)$ is defined by the full state of the world W, the agent effectively deals with a nonstationary problem [17, pp. 38–39]. That means, the same action in the same observed state does not always result in a converging reward value, given $t \to \infty$.

The agent's policy can be described as $\pi : V(CON) \to M$. Where CON is the set of contexts the agent is able to differentiate and M is the set of resulting motor activations. The agent learns the function $\delta^{con} : x_t \to x_{t+1}$ for each context *con*; goal is to learn the optimal function $\delta^{\hat{con}}$ that maximises the reward over all contexts *con* for $t \to \infty$.

We define the value function according to [13, p. 370] using $0 < \gamma < 1$ as discount in (1). Note, however, our value refers to past states—not future ones as [13, p. 370] originally proposes. The optimal policy is defined according to (2).

$$V^{\pi}(v_t) \equiv \sum_{i=0}^{t} \gamma^{t-i} r(v_i) \tag{1}$$

$$\pi^* \equiv \forall v.argmax_{\pi} V^{\pi}(v) \tag{2}$$

Sequence This definition of δ^{con} makes it obvious that the agent interacts with its environment optimally by recognising sequences of sensorimotor activations. Therefore the agent is able to anticipate the outcomes of its own actions as well as outside events.

Each situation or context *con* is an isolated component within the agent's memory graph structure $MEM \equiv V_{con} \times C$. Through temporal Hebbian learning [15] and local edge inhibition[2], this structure is adapted to the environment and to the actions of the agent. Coherent sequences form isolated graph components that can be regarded as abstract states of higher order. The process of generating those contexts can be retraced by considering algorithm 2.

Abstract Space The first layer of abstract representations is introduced if components of sensorimotor sequences are separated from the basic sensorimotor map. In general, abstract representations (or "contexts") in layer n are generated if sequences from level $n-1$ are separated by the removal of edges through local inhibition.

Each higher order state $con \in CON$ is structurally defined by a vector $\tilde{v}_{struc}$ describing its graph properties, just like each sensorimotor vector v is defined

[2] *Local inhibition* is motivated by neural inhibitory processes in the generation of visual contrast, compare [4].

by its activation. Those graph properties are converted to a vector through (3). Therefore, sequences of higher order states can be determined just like sequences of sensorimotor vectors (following algorithm 2). As this process is repeated, a hierarchical graph structure of more and more abstract cause and effect dependencies develops.

$$\tilde{v}_{struc} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, a_m = |A_m|, A_m = \text{set of nodes with degree } m \tag{3}$$

Structure & Function To overcome the problems of merely structural generalisation (vanishing intersections, no invariant representations, context dependency or in general: lacking symbolicity), the process of finding analogies in algorithm 2 needs to be extended. Because of temporal limitations, this shall only be sketched in the following.

Applying a pragmatic understanding of meaning, representations need to be evaluable concerning their functional role in a bigger frame or context (*zeugganzheit* according to [9, pp. 63–114]). Regarding the graph structure generated by the outlined procedure, one way of capturing functional aspects is by functional pattern recognition (FPR).

FPR applies the same process like algorithm 2, yet not on the content of the created representations, but on their edges. It pays no respect to the structural abstraction layer, instead it tries to form representations of sequences of frequently ocurring "embeddings". Therefore it generates a functional hierarchy independent from the existing structural hierarchy. The crucial aspect is, that not structural properties of active representations are observed, but rather their neighbourhood and the way the active representation is embedded within.

3.3 Implementation

Similarity $sim(v_1, v_2)$ is measured by (4) using Euclidean distance. According to (5), the evaluation of vectors $eval(v_t)$ is interpolated by referring to all nodes n_r in the current subgraph/context con that carry a reward $r(n)$.

To ignite interaction with the environment, the current evaluation $gradient_t$ is reduced by a constant $need$, according to (6). The variable $value_t$ is the weighted average, calculated by (7) and derived from (1)—accordingly s replaces γ.

$$sim(v_1, v_2) = \frac{1}{1 + dist(v_1, v_2)} \tag{4}$$

$$eval(v_t) = \sum_{n=n_r}^{con} sim(v_t, \text{vector of } n) * r(n) \tag{5}$$

$$gradient_t = \frac{1}{2} * (eval(v_t) - eval(v_{t-1})) - need \tag{6}$$

$$value_t = \frac{1}{s} * (gradient_t + value_{t-1} * (s-1)) \tag{7}$$

Algorithm 1: vectorOptimisation

Input: sensorimotor vector v_t, reward $r = [-1, 1] \in \mathbb{Z}$
Output: sensorimotor vector v_{t+1}

$context \leftarrow$ get parent of $node_{t-1}$;
$eval_t \leftarrow$ according to (5);
$value_t \leftarrow$ according to (7);
if $(r \neq 0 \wedge$ reward in $node_t \neq r) \vee (value_t < 0)$ **then**
  $node_t \leftarrow$ **new** node from v_t and r;
  add $node_t$ to $context$;
  put edge from $node_{t-1}$ to $node_t$ in $context$;
  $value_t \leftarrow 1$;
else $node_t \leftarrow findNextNode(v_{t-1}, v_t, context)$;
optimisation of vector in $node_{t-1}$;
$v_t \leftarrow$ get vector from $node_t$;
randomise v_t by $eval_t$;
put v_t in $node_t$;
return $node_t$

4 Results

Developing a first prototype, testing and debugging turned out to be quite difficult due to the plethora of data structures being generated. The implementation of Mountcastle's demand for a basic algorithm, that is able to express the multitude of mammal-level cognitive phenomena by the same basic algorithm, proved itself best to be tested within a simulated environment.

Test runs designated crucial parameters. Among those, the most relevant ones are *inhibition strength* and *degree of randomisation.*

Inhibition strength determines by which amount inhibited connections are reduced and whether reduction happens e. g. in a linear fashion or according to a sigmoid function. Strong local inhibition results in relatively small components, while weak inhibition allows the graph to grow before fragments stark to crumble away. As soon as the agent's memory reaches a size that makes efficient real time calculations hard, this parameter will be the first candidate for scaling measures. Most of run time complexity takes place during node search in lines 8–9 in (2). This process can be accelerated significantly if the number of nodes within each component can be reduced.

Algorithm 2: findNextNode

```
Input: vector v_{t-1}, vector v_t, node context
Output: node n_t

node_{t-1} ← find closest node to v_{t-1} in context;
node_t ← find closest node to v_t in context;
edge ← get edge from node_{t-1} to node_t in context;
if edge == nil then
    grandparent ← get parent of context;
    parentset ← get all child nodes of grandparent except context;
    foreach node_parent ∈ parentset do
        node_{t-1} ← find closest node to v_{t-1} in node_parent;
        node_t ← find closest node to v_t in node_parent;
        edge ← get edge from node_{t-1} to node_t in node_parent;
        v_parent ← get vector from node_parent;
        if edge ≠ nil then
            v_context ← get vector from context;
            context ← findNextNode(v_context, v_parent, grandparent);
            break;
        end
    end
    edge ← get edge from node_{t-1} to node_t in context;
    if edge == nil then put edge from node_{t-1} to node_t in context;
else reinforce edge;
inhibit alternative edges;
check for new components in context;
return node_t
```

Degree of randomisation controls the agent's explorativeness. Higher values enable reaching local optima faster, while with lower values, the chance of overshooting is significantly lower. Overshooting bears the additional risk in our case, that nodes, which might be exploited further, are left in favor of exploring the potential of n_{t+1}. This happens, because a reduction in $value_t$ is interpreted as indicator for reaching a local optimum.
Another question is, whether the degree of randomisation should be made dependant from $value_t$. Explorative behaviour during punishment enables leaving bad situations and entering better ones. In those situations on the other hand exploration should be minimised to reduce the chance of an unexpected negative turn of events.

5 Conclusion & Outlook

To extend the algorithm for covering functional sequence recognition, we need to determine where it engages in the outlined procedure. A possible contact point is line 19 in (2). At this point, it can be the case, that a similar context could be found, but it does not hold the sought-after transition from $node_{t-1}$ to $node_t$. If this unanticipated transition occurs, in the present solution we simply create the transition.

The advantage of this is, that over time, we get similar contexts to in fact hold similar transitions. Note that this is not guaranteed by similarity as we defined it in (3). However the disadvantage is, that excessive creation of edges might cause early representations to develop an average node degree that is too high to ensure reliable comparison. High dimensions in $\tilde{v}_{struc}$ would be overrepresented and tend to obliterate low dimensions holding significantly smaller values.

One possible solution to this is to refer to functional patterns that have been collected beforehand in case an expected transition is not to be found. This additional sequence recognition process receives informations like e. g. entropy, maximum, minimum or different average strengths of incoming and outgoing edges from the active node (v or $\tilde{v}$). This data is delivered independent from the active layer, such that functional patterns *beyond* one definite abstraction level can be determined and applied. In case line 4 in (2) is tested positive, functional sequence recognition can allocate an alternative node n_t *in the current context*, whose functional properties are frequently experienced as successive to the functional properties of n_{n-1}—again, *independent from context.* As has been stated before, this kind context-independency is regarded as a crucial necessity for synthetic symbolicity/semiotics.

Another way to implement functional sequence recognition within the presented framework is by replacing the structural abstraction hierarchy altogether by a functional hierarchy. New nodes might still be introduced like (1) proposes. But comparisons in (2) could abandon comparisons along $\tilde{v}_{struc}$ in favor of a yet to be defined $\tilde{v}_{func}$. As soon as this definition can be given, the existing framework needs to be modified only slighty, because the sequence recognition process is able to operate on any type of vectors—no matter if structural or functional. However, this implies grave epistemological presumptions, as the agent's world needs to be assembled in such a way that perceived structure at least partially defines applicable function. This assumption needs to be accepted axiomatically.

For implementing this extensions, the impact on the existing framework needs to be evaluated thoroughly. Because of the recursive character and the multitude of calculations executed, outcome cannot easily be forseen. Extensive testing of the basic version at hand and a capacious theoretical altercation is required. Yet the structural pattern recognition process presented so far gives no reason to believe that it is not perfectly suited to capture functional patterns just like structural patterns.

References

1. Anderson, J., Bothell, D., Byrne, M., Douglass, S., Lebiere, C., Qin, Y., et al.: An integrated theory of the mind. Psychological review 111(4), 1036–1060 (2004)
2. Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. ACM Computing Surveys (CSUR) 23(3), 345–405 (1991)
3. Baum, E.: What is thought? Bradford Books, MIT Press (2004)
4. Békésy, G.: Sensory Inhibition. Princeton paperbacks, Princeton University Press (1967)
5. Gyftodimos, E., Flach, P.: Hierarchical bayesian networks: A probabilistic reasoning model for structured domains. In: Proceedings of the ICML-2002 Workshop on Development of Representations. University of New South Wales. Citeseer (2002)
6. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena 42(1-3), 335–346 (1990)
7. Hawkins, J., Blakeslee, S.: On intelligence. Owl Books (2005)
8. Hawkins, J., George, D., Niemasik, J.: Sequence memory for prediction, inference and behaviour. Philosophical Transactions of the Royal Society B: Biological Sciences 364(1521), 1203 (2009)
9. Heidegger, M.: Sein und Zeit. Max Niemeyer Verlag (2001)
10. Laird, J.: Extending the Soar cognitive architecture. In: Proceeding of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference. pp. 224–235. IOS Press (2008)
11. Lazkano, E., Sierra, B., Astigarraga, A., Martínez-Otzeta, J.: On the use of Bayesian Networks to develop behaviours for mobile robots. Robotics and Autonomous Systems 55(3), 253–265 (2007)
12. Markram, H.: The blue brain project. Nature Reviews Neuroscience 7(2), 153–160 (2006)
13. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
14. Mountcastle, V.: An organizing principle for cerebral function: the unit model and the distributed system. In: Edelman, G., Mountcastle, V. (eds.) The Mindful Brain. MIT Press, Cambridge, Mass. (1978)
15. Porr, B., Wörgötter, F.: Temporal hebbian learning in rate-coded neural networks: A theoretical approach towards classical conditioning. Artificial Neural Networks—ICANN 2001 2130, 1115–1120 (2001)
16. Sun, R.: A tutorial on CLARION 5.0. Retrieved on October 15, 2003 (2003)
17. Sutton, R., Barto, A.: Reinforcement learning: an introduction. Adaptive computation and machine learning, MIT Press (1998)