The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

# *k*-th Order Intelligences: Learning To Learn To Do

Francisco J. Arjonilla[*1] Yuichi Kobayashi[*1]

[*1] Graduate School of Science and Technology, Shizuoka University

We propose a novel classification of intelligence based on distinguishing model exploitation from model exploration in order to improve our general understanding of intelligence and its limitations. For this purpose, we define computational problems by traditional function execution, which implicitly hold the model of the problem to solve, and learning problems by the meta-methods that produce computational methods. Learning problems are then assimilated to computational methods which hold implicit meta-models. The process is repeated iteratively, with each iteration named a *k*-th order intelligence. However, we show that the infinite sequence of classes of intelligence that emerges poses difficulties for meta-model exploration. We suggest using self-referential meta-models to break the escalation of orders, and we introduce some of the problems associated to this approach.

## 1. Introduction

The definition of intelligence is a highly debated topic that finds no consensus amongst researchers [Legg 2007]. In general, papers that tackle intelligence begin by offering a definition that is convenient for the topics dealt with and often matches the subjective view of the author on intelligence. Many of these definitions are too abstract or measure single capabilities. Intelligence includes concepts such as learning, exploration vs. exploitation, algorithms, knowledge representation, pattern recognition and many others. With such a broad scope, it is difficult to reconcile all aspects of intelligence in a single line of research for the purpose of unifying efforts into developing a plausible general theory of intelligence, hopefully facilitating collaborations and merging researches.

We propose a novel classification of Artificial Intelligence based on computational problems vs. learning problems that builds up from traditional algorithmic methods to general meta-models. In the next section we characterize computational problems and continue with learning problems on the following section. We then consider a different boundary for the learning agent such that we can equalize learning problems to computational problems, thus resulting in a hierarchical specification of meta-learning methods that emerge by induction. Lastly, we show how this hierarchy leads to fundamental problems to achieve General Intelligence.

## 2. Computational Problems

We start the discussion by defining computational problems as those problems solved by traditional function execution, where typically input data is processed and transformed to produce output data. The function realizes a fixed algorithm that holds an implicit model of the problem and the instructions to solve it. Hereupon, we will refer to these functions and problems as *first order intelligence* and *first order problems*, respectively. Typical examples are those informally referred as narrow intelligence, which include symbolic processing, calculation tasks, control algorithms and basically any function that yields output data. They are characterized by immutable algorithms that are specialized in concrete tasks. With respect to agent-environment systems, first order intelligence encompasses all the methods that imply exploitation of a model and enable an agent to interact and make changes to the environment, such as trained neural networks, expert systems, reactive systems and natural language processing, to name a few. Intuitively, first order intelligence is understood as *doing*.

## 3. Learning Problems

In contrast, learning problems are differentiated by the scope of the transformation. Rather than making changes to the environment, a learning process modifies the methods used in computational problems. Therefore, the target of a learning process is not the environment, but the agent itself. Learning problems arise when first order intelligence is unable to cope with computational problems due to a lack of an appropriate model [Unruh 1989]. In that case, learning processes generate candidate new models, *i.e.* functions, and test them against computational problems. We call these processes *second order intelligence*. Whereas first order intelligence exploits models, second order intelligence explores the model space. Representative examples include Bayesian networks, evolutionary programming and reinforcement learning. The common characteristic amongst these methods is that the output is a function, code listing or model, such that their input and output correspond to the ones that characterize first order intelligence, *i.e.* raw data. The functions produced are often defined by algorithmically choosing the appropriate weights. That is the case with neural network training algorithms such as gradient descent [Werbos 1990]. During training, the neural

Contact: Francisco J. Arjonilla (Paco), Graduate School of Science and Technology, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu City, +81 (0) 53-478-1604, pacoarjonilla@yahoo.es

network weights are adjusted until the measured error on the training data falls below a certain threshold, which represents the acceptance level of the neural network. This corresponds to second order intelligence because training does not affect the environment, but the resultant neural network does. After training, the neural network holds an implicit model of the training data in its weights, as characterized by first order processes. As opposed to doing, second order intelligence is understood intuitively as *learning to do*.

### 3.1 Learning Problems as Computational Problems

Let us now change the perspective on learning problems by redefining the boundaries of the agent by relocating the realization of first order intelligence to the environment. In other words, we consider the computing device, or at least the memory space reserved for first order processes, as external to our agent. Direct control of the environment is hence reduced to controlling what is executed in the first order computing device, which in turn controls the rest of the environment. The new boundaries redefine computational problems and learning problems as stated in the previous sections, yet this agent's computational processes hold meta-models, or models of models. Thus, learning problems for first order intelligence are the same as computational problems for second order intelligence. That is to say, learning problems are computational meta-problems in the domain of computational problems; exploration of a model is exploitation of a meta-model.

## 4. Higher Orders of Intelligence

In the previous sections we have treated first and second order intelligences, and how a second order intelligence is interpreted as a first order meta-intelligence. We now repeat the same process to arrive at third order intelligence from second order intelligence. In the same way that second order intelligence takes over first order intelligence when there are no models to cope with a problem, third order intelligence triggers when there are no meta-models available [Schmidhuber 2005]. However, third order processes are rarely found in Artificial Intelligence. One example is [Naik 1992] where a meta-neural network establishes the training parameters of a basic neural network, improving learning rates for solving problems that are similar to previously solved ones. Intuitively, third order intelligence is regarded as *learning to learn* (to do).

The next natural step is to consider higher order intelligences. These orders emerge when models, meta-models, meta-meta-models, … get exhausted. Hence, an infinite sequence of orders develops where exploration of a meta-model in the $k$-th order is exploitation of the meta-model in the $(k+1)$-th order [Turing 1939]. Unfortunately, higher orders are more complex, abstract and difficult to visualize intuitively: *learning to learn to learn to* … In A.I., there are no known methods that take the role of higher order intelligences. On the contrary, this role is taken by human researchers, who generally propose new methods corresponding to first order intelligence (*e.g.* GOFAI) or second/third order intelligence (*e.g.* machine learning). When the methods in A.I. fail for a given problem at some order, researchers ultimately perform exploration of the meta-models.

### 4.1 Breaking the Escalation of Orders

In order to overcome the lack of higher order processes, we propose a general meta-model that consists of a self-referential method that can modify itself. This way, the meta-model of this method is the model itself, effectively discontinuing the trend towards infinite orders by equalizing $k$-th order and $(k+1)$-th order for some $k$. Such a method is canonically realized by a function whose input and output data are the binary encoding of the function itself. Nevertheless, a self-referential function that can explore its own model conflicts with well-known problems of axiomatic systems related to incompleteness [Gödel 1931], inconsistency and the P *vs.* NP problem [Cook 1971].

## References

[Cook 1971] Cook, S. A., *The Complexity of Theorem-Proving Procedures*, Proceedings of the third annual ACM symposium on Theory of Computing, ACM Press, 1971.

[Gödel 1931] Gödel, K., *Über formal Unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, Monatshefte Für Mathematik Und Physik, 1939.

[Hutter 2001] Hutter, M., *Towards a Universal Theory of Artificial Intelligence based on Algorithmic Probability and Sequential Decisions*, Proceedings of the 12th European Conference on Machine Learning, 2001.

[Legg 2007] Legg, S., & Hutter, M., *Universal Intelligence: A Definition of Machine Intelligence*, Minds and Machines, 2007.

[Naik 1992] Naik, D. K., & Mammone, R. J., *Meta-neural networks that learn by learning,* International Joint Conference on Neural Networks, IEEE, 1992.

[Pfeifer 2001] Pfeifer, R., & Scheier, C., *Understanding intelligence,* The MIT Press, 2001.

[Schmidhuber 2005] Schmidhuber, J., *Gödel Machines: Self-Referential Universal Problem Solvers Making Provably Optimal Self-Improvements*, Artificial General Intelligence, Springer, 2005

[Sternberg 1977] Sternberg, R. J, *Intelligence, Information Processing, and Analogical Reasoning*, John Wiley & Sons, 1997.

[Turing 1939] Turing, A. M., *Systems of Logic Based on Ordinals*, Proceedings of the London Mathematical Society, 1939.

[Unruh 1989] Unruh, A., & Rosenbloom, P. S., *Abstraction in Problem Solving and Learning*, Proceedings of the 11th International Joint Conference on Artificial Intelligence, 1989.

[Werbos 1990] Werbos, P. J., *Backpropagation Through Time: What It Does and How to Do It*, Proceedings of the IEEE, 1990.