The Optimal Choice of Hypothesis Is the Weakest, Not the Shortest

Michael Timothy Bennett(✉)

The Australian National University, Canberra, Australia
michael.bennett@anu.edu.au
http://www.michaeltimothybennett.com/

Abstract. If A and B are sets such that $A \subset B$, generalisation may be understood as the inference from A of a hypothesis sufficient to construct B. One might infer any number of hypotheses from A, yet only some of those may generalise to B. How can one know which are likely to generalise? One strategy is to choose the shortest, equating the ability to compress information with the ability to generalise (a "proxy for intelligence"). We examine this in the context of a mathematical formalism of enactive cognition. We show that compression is neither necessary nor sufficient to maximise performance (measured in terms of the probability of a hypothesis generalising). We formulate a proxy unrelated to length or simplicity, called weakness. We show that if tasks are uniformly distributed, then there is no choice of proxy that performs at least as well as weakness maximisation in all tasks while performing strictly better in at least one. In experiments comparing maximum weakness and minimum description length in the context of binary arithmetic, the former generalised at between 1.1 and 5 times the rate of the latter. We argue this demonstrates that weakness is a far better proxy, and explains why Deepmind's Apperception Engine is able to generalise effectively.

Keywords: simplicity · induction · artificial general intelligence

1 Introduction

If A and B are sets such that $A \subset B$, generalisation may be understood as the inference from A of a hypothesis sufficient to construct B. One might infer any number of hypotheses from A, yet only some of those may generalise to B. How can one know which are likely to generalise? According to Ockham's Razor, the simpler of two explanations is the more likely [2]. Simplicity is not itself a measurable property, so the minimum description length principle [3] relates simplicity to length. Shorter representations are considered to be simpler, and tend to generalise more effectively. This is often applied in the context of induction by comparing the length of programs that explain what is observed (to chose the shortest, all else being equal). The ability to identify shorter representations is compression, and the ability to generalise is arguably intelligence [4]. Hence the

P. Hammer et al. (Eds.): AGI 2023, LNAI 13921, pp. 42–51, 2023.
https://doi.org/10.1007/978-3-031-33469-6_5

ability to compress information is often portrayed as a proxy for intelligence [5], even serving as the foundation [6–8] of the theoretical super-intelligence AIXI [9]. That compression is a good proxy seems to have gone unchallenged. The optimal choice of hypothesis is widely considered to be the shortest. We show that it is not[1]. We present an alternative, unrelated to description length, called weakness. We prove that to maximise the probability that one's hypotheses generalise, it is necessary and sufficient to infer the weakest valid hypotheses possible[2].

2 Background Definitions

To do so, we employ a formalism of enactive cognition [1,10,11], in which sets of declarative programs are related to one another in such a way as to form a lattice. This unusual representation is necessary to ensure that both the weakness and description length of a hypothesis are well defined[3]. This formalism can be understood in three steps.

1. The environment is represented as a set of declarative programs.
2. A finite subset of the environment is used to define a language with which to write statements that behave as logical formulae.
3. Finally, induction is formalised in terms of tasks made up of these statements.

Definition 1 (environment)

- *We assume a set Φ whose elements we call* ***states****, one of which we single out as the* ***present state***[4].
- *A* ***declarative program*** *is a function $f : \Phi \rightarrow \{true, false\}$, and we write P for the set of all declarative programs. By an* ***objective truth*** *about a state ϕ, we mean a declarative program f such that $f(\phi) = true$.*

Definition 2 (implementable language)

- *$\mathfrak{V} = \{V \subset P : V \textit{ is finite}\}$ is a set whose elements we call* ***vocabularies****, one of which we single out as* ***the vocabulary*** *$\mathfrak{v}$ for an implementable language.*
- *$L_\mathfrak{v} = \{l \subseteq \mathfrak{v} : \exists \phi \in \Phi \; (\forall p \in l : p(\phi) = true)\}$ is a set whose elements we call* ***statements***[5]. *$L_\mathfrak{v}$ follows from Φ and $\mathfrak{v}$. We call $L_\mathfrak{v}$ an* ***implementable language****.*

[1] This proof is conditional upon certain assumptions regarding the nature of cognition as enactive, and a formalism thereof.

[2] Assuming tasks are uniformly distributed, and weakness is well defined.

[3] An example of how one might translate propositional logic into this representation is given at the end of this paper. It is worth noting that this representation of logical formulae addresses the symbol grounding problem [12], and was specifically constructed to address subjective performance claims in the context of AIXI [13].

[4] Each state is just reality from the perspective of a point along one or more dimensions. States of reality must be separated by something, or there would be only one state of reality. For example two different states of reality may be reality from the perspective of two different points in time, or in space and so on.

[5] Statements are the logical formulae about which we will reason.

- *$l \in L_{\mathfrak{v}}$ is **true** iff the present state is ϕ and $\forall p \in l : p(\phi) = true$.*
- *The **extension of a statement** $a \in L_{\mathfrak{v}}$ is $Z_a = \{b \in L_{\mathfrak{v}} : a \subseteq b\}$.*
- *The **extension of a set of statements** $A \subseteq L_{\mathfrak{v}}$ is $Z_A = \bigcup\limits_{a \in A} Z_a$.*

(Notation). *Z with a subscript is the extension of the subscript*[6]*. Lower case letters represent statements, and upper case represent sets of statements.*

Definition 3 ($\mathfrak{v}$-task). *For a chosen $\mathfrak{v}$, a task α is $\langle S_\alpha, D_\alpha, M_\alpha \rangle$ where:*

- *$S_\alpha \subset L_{\mathfrak{v}}$ is a set whose elements we call **situations** of α.*
- *S_α has the extension Z_{S_α}, whose elements we call **decisions** of α.*
- *$D_\alpha = \{z \in Z_{S_\alpha} : z \text{ is correct}\}$ is the set of all decisions which complete α.*
- *$M_\alpha = \{l \in L_{\mathfrak{v}} : Z_{S_\alpha} \cap Z_l = D_\alpha\}$ whose elements we call **models** of α.*

$\Gamma_{\mathfrak{v}}$ is the set of all tasks[7]*.*

(Notation). *If $\omega \in \Gamma_{\mathfrak{v}}$, then we will use subscript ω to signify parts of ω, meaning one should assume $\omega = \langle S_\omega, D_\omega, M_\omega \rangle$ even if that isn't written.*

(How a task is completed). *Assume we've a $\mathfrak{v}$-task ω and a hypothesis $\boldsymbol{h} \in L_{\mathfrak{v}}$ s.t.*

1. *we are presented with a situation $s \in S_\omega$, and*
2. *we must select a decision $z \in Z_s \cap Z_{\boldsymbol{h}}$.*
3. *If $z \in D_\omega$, then z is correct and the task is complete. This occurs if $\boldsymbol{h} \in M_\omega$.*

3 Formalising Induction

Definition 4 (probability). *We assume a uniform distribution over $\Gamma_{\mathfrak{v}}$.*

Definition 5 (generalisation). *A statement l generalises to $\alpha \in \Gamma_{\mathfrak{v}}$ iff $l \in M_\alpha$. We say l generalises from α to $\mathfrak{v}$-task ω if we first obtain l from M_α and then find it generalises to ω.*

Definition 6 (child and parent). *A $\mathfrak{v}$-task α is a child of $\mathfrak{v}$-task ω if $S_\alpha \subset S_\omega$ and $D_\alpha \subseteq D_\omega$. This is written as $\alpha \sqsubset \omega$. If $\alpha \sqsubset \omega$ then ω is then a parent of α.*

A proxy is meant to estimate one thing by measuring another. In this case, if intelligence is the ability to generalise [4,10], then a greater proxy value is meant to indicate that a statement is more likely to generalise. Not all proxies are effective (most will be useless). We focus on two in particular.

[6] e.g. Z_s is the extension of s.

[7] For example, we might represent chess as a supervised learning problem where $s \in S_\alpha$ is the state of a chessboard, $z \in Z_s$ is a sequence of moves by two players that begins in s, and $d \in D_\alpha \cap Z_s$ is such a sequence of moves that terminates in victory for one player in particular (the one undertaking the task).

Definition 7 (proxy for intelligence). *A proxy is a function parameterized by a choice of* $\mathfrak{v}$ *such that* $q_\mathfrak{v} : L_\mathfrak{v} \rightarrow \mathbb{N}$*. The set of all proxies is* Q*.*

(Weakness). *The weakness of a statement* l *is the cardinality of its extension* $|Z_l|$*. There exists* $q_\mathfrak{v} \in Q$ *such that* $q_\mathfrak{v}(l) = |Z_l|$*.*

(Description length). *The description length of a statement* l *is its cardinality* $|l|$*. Longer logical formulae are considered less likely to generalise [3], and a proxy is something to be maximised, so description length as a proxy is* $q_\mathfrak{v} \in Q$ *such that* $q_\mathfrak{v}(l) = \frac{1}{|l|}$*.*

A child task may serve as an ostensive definition [14] of its parent, meaning one can generalise from child to parent.

Definition 8 (induction). α *and* ω *are* $\mathfrak{v}$*-tasks such that* $\alpha \sqsubset \omega$*. Assume we are given a proxy* $q_\mathfrak{v} \in Q$*, the complete definition of* α *and the knowledge that* $\alpha \sqsubset \omega$*. We are not given the definition of* ω*. The process of induction would proceed as follows:*

1. *Obtain a hypothesis by computing a model* $\mathbf{h} \in \underset{m \in M_\alpha}{\arg\max}\ q_\mathfrak{v}(m)$*.*
2. *If* $\mathbf{h} \in M_\omega$*, then we have generalised from* α *to* ω*.*

4 Proofs

Proposition 1 (sufficiency). *Weakness is a proxy sufficient to maximise the probability that induction generalises from* α *to* ω*.*

Proof: You're given the definition of $\mathfrak{v}$-task α from which you infer a hypothesis $\mathbf{h} \in M_\alpha$. $\mathfrak{v}$-task ω is a parent of α to which we wish to generalise:

1. The set of statements which *might* be decisions addressing situations in S_ω and not S_α, is $\overline{Z_{S_\alpha}} = \{l \in L_\mathfrak{v} : l \notin Z_{S_\alpha}\}$.
2. For any given $\mathbf{h} \in M_\alpha$, the extension $Z_\mathbf{h}$ of $\mathbf{h}$ is the set of decisions $\mathbf{h}$ implies. The subset of $Z_\mathbf{h}$ which fall outside the scope of what is required for the known task α is $\overline{Z_{S_\alpha}} \cap Z_\mathbf{h}$ (because Z_{S_α} is the set of all decisions we might make when attempting α, and so the set of all decisions that can't be made when undertaking α is $\overline{Z_{S_\alpha}}$ because those decisions occur in situations that aren't part of S_α).
3. $|\overline{Z_{S_\alpha}} \cap Z_\mathbf{h}|$ increases monotonically with $|Z_\mathbf{h}|$, because $\forall z \in Z_m : z \notin \overline{Z_{S_\alpha}} \rightarrow z \in Z_{S_\alpha}$.
4. $2^{|\overline{Z_{S_\alpha}}|}$ is the number of tasks which fall outside of what it is necessary for a model of α to generalise to (this is just the powerset of $\overline{Z_{S_\alpha}}$ defined in step 2), and $2^{|\overline{Z_{S_\alpha}} \cap Z_\mathbf{h}|}$ is the number of those tasks to which a given $\mathbf{h} \in M_\alpha$ does generalise.

5. Therefore the probability that a given model $\mathbf{h} \in M_\alpha$ generalises to the unknown parent task ω is

$$p(\mathbf{h} \in M_\omega \mid \mathbf{h} \in M_\alpha, \alpha \sqsubset \omega) = \frac{2^{|\overline{Z_{S_\alpha}} \cap Z_{\mathbf{h}}|}}{2^{|\overline{Z_{S_\alpha}}|}}$$

$p(\mathbf{h} \in M_\omega \mid \mathbf{h} \in M_\alpha, \alpha \sqsubset \omega)$ is maximised when $|Z_{\mathbf{h}}|$ is maximised.

Proposition 2 (necessity). *To maximise the probability that induction generalises from α to ω, it is necessary to use weakness as a proxy, or a function thereof*[8]*.*

Proof: Let α and ω be defined exactly as they were in the proof of Proposition 1.

1. If $\mathbf{h} \in M_\alpha$ and $Z_{S_\omega} \cap Z_{\mathbf{h}} = D_\omega$, then it must be he case that $D_\omega \subseteq Z_{\mathbf{h}}$.
2. If $|Z_{\mathbf{h}}| < |D_\omega|$ then generalisation cannot occur, because that would mean that $D_\omega \not\subseteq Z_{\mathbf{h}}$.
3. Therefore generalisation is only possible if $|Z_m| \geq |D_\omega|$, meaning a sufficiently weak hypothesis is necessary to generalise from child to parent.
4. The probability that $|Z_m| \geq |D_\omega|$ is maximised when $|Z_m|$ is maximised. Therefore to maximise the probability induction results in generalisation, it is necessary to select the weakest hypothesis.

To select the weakest hypothesis, it is necessary to use weakness (or a function thereof) as a proxy.

Remark 1 (prior). The above describes inference from a child to a parent. However, it follows that increasing the weakness of a statement increases the probability that it will generalise to any task (not just a parent of some given child). As tasks are uniformly distributed, every statement in $L_{\mathfrak{v}}$ is a model to one or more tasks, and the number of tasks to which each statement $l \in L_{\mathfrak{v}}$ generalises is $2^{|Z_l|}$. Hence the probability of generalisation[9] to ω is $p(\mathbf{h} \in M_\omega \mid \mathbf{h} \in L_{\mathfrak{v}}) = \frac{2^{|Z_{\mathbf{h}}|}}{2^{|L_{\mathfrak{v}}|}}$. This assigns a probability to every statement $l \in L_{\mathfrak{v}}$ given an implementable language. It is a probability distribution in the sense that the probability of mutually exclusive statements sums to one[10]. This prior may be considered universal in the very limited sense that it assigns a probability to every conceivable hypothesis (where what is conceivable depends upon the implementable language) absent any parameters or specific assumptions about the task as with AIXI's intelligence order relation [9, def. 5.14 pp. 147][11]. As the vocabulary $\mathfrak{v}$ is finite, $L_{\mathfrak{v}}$ must also be finite, and so p is computable.

[8] For example we might use weakness multiplied by a constant to the same effect.

[9] $\frac{2^{|Z_{\mathbf{h}}|}}{2^{|L_{\mathfrak{v}}|}}$ is maximised when $\mathbf{h} = \emptyset$, because the optimal hypothesis given no information is to assume nothing (you've no sequence to predict, so why make assertions that might contradict the environment?).

[10] Two statements a and b are mutually exclusive if $a \notin Z_b$ and $b \notin Z_a$, which we'll write as $\mu(a, b)$. Given $x \in L_{\mathfrak{v}}$, the set of all mutually exclusive statements is a set $K_x \subset L_{\mathfrak{v}}$ such that $x \in K_x$ and $\forall a, b \in K_x : \mu(a, b)$. It follows that $\forall x \in L_{\mathfrak{v}}, \sum_{b \in K_x} p(b) = 1$.

[11] We acknowledge that some may object to the term universal, because $\mathfrak{v}$ is finite.

We have shown that, if tasks are uniformly distributed, then weakness is a necessary and sufficient proxy to maximise the probability that induction generalises. It is important to note that another proxy may perform better given cherry-picked combinations of child and parent task for which that proxy is suitable. However, such a proxy would necessarily perform worse given the uniform distribution of all tasks. Can the same be said of description length?

Proposition 3. *Description length is neither a necessary nor sufficient proxy for the purposes of maximising the probability that induction generalises.*

Proof: In Propositions 1 and 2 we proved that weakness is a necessary and sufficient choice of proxy to maximise the probability of generalisation. It follows that either maximising $\frac{1}{|m|}$ (minimising description length) maximises $|Z_m|$ (weakness), or minimisation of description length is unnecessary to maximise the probability of generalisation. Assume the former, and we'll construct a counterexample with $\mathfrak{v} = \{a, b, c, d, e, f, g, h, j, k, z\}$ s.t. $L_\mathfrak{v} = \{\{a, b, c, d, j, k, z\}, \{e, b, c, d, k\}, \{a, f, c, d, j\}, \{e, b, g, d, j, k, z\}, \{a, f, c, h, j, k\}, \{e, f, g, h, j, k\}\}$ and a task α where

- $S_\alpha = \{\{a, b\}, \{e, b\}\}$
- $D_\alpha = \{\{a, b, c, d, j, k, z\}, \{e, b, g, d, j, k, z\}\}$
- $M_\alpha = \{\{z\}, \{j, k\}\}$

Weakness as a proxy selects $\{j, k\}$, while description length as a proxy selects $\{z\}$. This demonstrates the minimising description length does not necessarily maximise weakness, and maximising weakness does not minimise description length. As weakness is necessary and sufficient to maximise the probability of generalisation, it follows that minimising description length is neither.

5 Experiments

Included with this paper is a Python script to perform two experiments using PyTorch with CUDA, SymPy and A^* [15–18] (see technical appendix for details). In these two experiments, a toy program computes models to 8-bit string prediction tasks (binary addition and multiplication). The purpose of these experiments was to compare weakness and description length as proxies.

5.1 Setup

To specify tasks with which the experiments would be conducted, we needed a vocabulary $\mathfrak{v}$ with which to describe simple 8-bit string prediction problems. There were 256 states in Φ, one for every possible 8-bit string. The possible statements were then all the expressions regarding those 8 bits that could be written in propositional logic (the simple connectives $\neg$, $\wedge$ and $\vee$ needed to perform binary arithmetic – a written example of how propositional logic can be used in to specify $\mathfrak{v}$ is also included in the appendix). In other words, for each statement in $L_{\mathfrak{v}}$ there existed an equivalent expression in propositional logic. For efficiency, these statements were implemented as either PyTorch tensors or SymPy expressions in different parts of the program, and converted back and forth as needed (basic set and logical operations on these propositional tensor representations were implemented for the same reason). A $\mathfrak{v}$-task was specified by choosing $D_n \subset L_{\mathfrak{v}}$ such that all $d \in D_n$ conformed to the rules of either binary addition or multiplication with 4-bits of input, followed by 4-bits of output.

5.2 Trials

Each experiment had parameters were "operation" and "number_of_trials". For each trial the number $|D_k|$ of examples ranged from 4 to 14. A trial had 2 phases.

Training Phase

1. A task n (referred to in code as T_n) was generated:
   (a) First, every possible 4-bit input for the chosen binary operation was used to generate an 8-bit string. These 16 strings then formed D_n.
   (b) A bit between 0 and 7 was then chosen, and S_n created by cloning D_n and deleting the chosen bit from every string (S_n contained 16 different 7-bit strings, each of which was a sub-string of an element of D_n).
2. A child-task $k = \langle S_k, D_k, M_k \rangle$ (referred to in code as T_k) was sampled (assuming a uniform distribution over children) from the parent task T_n. Recall, $|D_k|$ was determined as a parameter of the trial.
3. From T_k two models were then generated; a weakest c_w, and a MDL c_{mdl}.

Testing Phase: For each model $c \in \{c_w, c_{mdl}\}$, the testing phase was as follows:

1. The extension Z_c of c was then generated.
2. A prediction D_{recon} was made s.t. $D_{recon} = \{z \in Z_c : \exists s \in S_n \ (s \subset z)\}$.
3. D_{recon} was then compared to the ground truth D_n, and results recorded.

Between 75 and 256 trials were run for each value of the parameter $|D_k|$. Fewer trials were run for larger values of $|D_k|$ as these took longer to process. The results of these trails were then averaged for each value of $|D_k|$.

5.3 Results

Two sorts of measurements were taken for each trial. The first was **the rate at generalisation occurred**. Generalisation was deemed to have occurred where $D_{recon} = D_n$. The number of trials in which generalisation occurred was measured, and divided by n to obtain the rate of generalisation for c_w and c_{mdl}. Error was computed as a Wald 95% confidence interval. The second measurement was **the average extent to which models generalised**. Even where $D_{recon} \neq D_n$, the extent to which models generalised could be ascertained. $\frac{|D_{recon} \cap D_n|}{|D_n|}$ was measured and averaged for each value of $|D_k|$, and the standard error computed. The results (see Tables 1 and 2) demonstrate that weakness is a better proxy for intelligence than description length. The generalisation rate for c_w was between 110–500% of c_{mdl}, and the extent was between $103 - 156\%$.

Table 1. Results for Binary Addition

$\lvert D_k \rvert$	c_w				c_{mdl}			
	Rate	±95%	AvgExt	StdErr	Rate	±95%	AvgExt	StdErr
6	.11	.039	.75	.008	.10	.037	.48	.012
10	.27	.064	.91	.006	.13	.048	.69	.009
14	.68	.106	.98	.005	.24	.097	.91	.006

Table 2. Results for Binary Multiplication

$\lvert D_k \rvert$	c_w				c_{mdl}			
	Rate	±95%	AvgExt	StdErr	Rate	±95%	AvgExt	StdErr
6	.05	.026	.74	.009	.01	.011	.58	.011
10	.16	.045	.86	.006	.08	.034	.78	.008
14	.46	.061	.96	.003	.21	.050	.93	.003

6 Concluding Remarks

We have shown that, if tasks are uniformly distributed, then weakness maximisation is necessary and sufficient to maximise the probability that induction will produce a hypothesis that generalises. It follows that there is no choice of proxy that performs at least as well as weakness maximisation across all possible combinations of child and parent task while performing strictly better in at least one. We've also shown that the minimisation of description length is neither necessary nor sufficient. This calls into question the relationship between compression and intelligence [5,19,20], at least in the context of enactive cognition. This is supported by our experimental results, which demonstrate that weakness is a far better predictor of whether a hypothesis will generalise, than description length. Weakness should not be conflated with Ockham's Razor. A simple statement need not be weak, for example "all things are blue crabs". Likewise, a complex utterance can assert nothing. Weakness is a consequence of extension, not form. If weakness is to be understood as an epistemological razor, it is this (which we humbly suggest naming "Bennett's Razor"):

> *Explanations should be no more specific than necessary.*[12]

The Apperception Engine: The Apperception Engine [21–23] (Evans et al. of Deepmind) is an inference engine that generates hypotheses that generalise often. To achieve this, Evans formalised Kant's philosophy to give the engine a "strong inductive bias". The engine forms hypotheses from only very general

[12] We do not know which possibilities will eventuate. A less specific statement contradicts fewer possibilities. Of all hypotheses sufficient to explain what we perceive, the least specific is most likely.

assertions, meaning logical formulae which are universally quantified. That is possible because the engine uses language specifically tailored to efficiently represent the sort of sequences to which it is applied. Our results suggest a simpler and more general explanation of why the engine's hypotheses generalise so well. The tailoring of logical formulae to represent certain sequences amounts to a choice of $\mathfrak{v}$, and the use of only universally quantified logical formulae maximises the weakness of the resulting hypothesis. To apply this approach to induction from child $\mathfrak{v}$-task α to parent ω would mean we only entertain a model $m \in M_\alpha$ if $p(m \in M_\omega \mid m \in M_\alpha) = 1$. Obviously this can work well, but only for the subset of possible tasks that the vocabulary is able to describe in this way (anything else will not be able to be represented as a universally quantified rule, and so will not be represented at all [24]). This illustrates how future research may explore choices of $\mathfrak{v}$ in aid of more efficient induction in particular sorts of task, such as the inference of linguistic meaning and intent (see appendix).

Neural Networks: How might a task be represented in the context of conventional machine learning? Though we use continuous real values in base 10 to formalise neural networks, all computation still takes place in a discrete, finite and binary system. A finite number of imperative programs composed a finite number of times may be represented by a finite set of declarative programs. Likewise, activations within a network given an input can be represented as a finite set of declarative programs, expressing a decision. The choice of architecture specifies the vocabulary in which this is written, determining what sort of relations can be described according to the Chomsky Hierarchy [25]. The reason why LLMs are so prone to fabrication and inconsistency may be because they are optimised only to minimise loss, rather than maximise weakness [10]. Perhaps grokking [26] can be induced by optimising for weakness. Future research should investigate means by which weakness can be maximised in the context of neural networks.

Acknowledgement. Appendices available on GitHub [1], supported by JST (JPMJMS2033).

References

1. Bennett, M.T.: Technical Appendices. Version 1.2.1 (2023). https://doi.org/10.5281/zenodo.7641742. https://github.com/ViscousLemming/Technical-Appendices
2. Sober, E.: Ockham's Razors: A User's Manual. Cambridge University Press (2015)
3. Rissanen, J.: Modeling by shortest data description*. Automatica **14**, 465–471 (1978)
4. Chollet, F.: On the Measure of Intelligence (2019)
5. Chaitin, G.: The limits of reason. Sci. Am. **294**(3), 74–81 (2006)
6. Solomonoff, R.: A formal theory of inductive inference. Part I. Inf. Control **7**(1), 1–22 (1964)

7. Solomonoff, R.: A formal theory of inductive inference. Part II. Inf. Control **7**(2), 224–254 (1964)
8. Kolmogorov, A.: On tables of random numbers. Sankhya: Indian J. Stati. A 369–376 (1963)
9. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability. Springer, Heidelberg (2010)
10. Bennett, M.T.: Symbol emergence and the solutions to any task. In: Goertzel, B., Iklé, M., Potapov, A. (eds.) AGI 2021. LNCS (LNAI), vol. 13154, pp. 30–40. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-93758-4_4
11. Ward, D., Silverman, D., Villalobos, M.: Introduction: the varieties of enactivism. Topoi **36**(3), 365–375 (2017). https://doi.org/10.1007/s11245-017-9484-6
12. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena **42**(1), 335–346 (1990)
13. Leike, J., Hutter, M.: Bad universal priors and notions of optimality. In: Proceedings of the 28th COLT, PMLR, pp. 1244–1259 (2015)
14. Gupta, A.: Definitions. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Winter 2021. Stanford University (2021)
15. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS. Curran Association Inc., USA (2019)
16. Kirk, D.: NVIDIA Cuda Software and GPU parallel computing architecture. In: ISMM 2007, Canada, pp. 103–104. ACM (2007)
17. Meurer, A., et al.: SymPy: symbolic computing in Python. PeerJ Comput. Sci. **3**, e103 (2017). https://doi.org/10.7717/peerj-cs.103
18. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
19. Hernández-Orallo, J., Dowe, D.L.: Measuring universal intelligence: towards an anytime intelligence test. Artif. Intell. **174**(18), 1508–1539 (2010)
20. Legg, S., Veness, J.: An approximation of the universal intelligence measure. In: Dowe, D.L. (ed.) Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence. LNCS, vol. 7070, pp. 236–249. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-44958-1_18
21. Evans, R.: Kant's cognitive architecture. Ph.D. thesis. Imperial (2020)
22. Evans, R., Sergot, M., Stephenson, A.: Formalizing Kant's rules. J. Philos. Logic **49**, 613–680 (2020)
23. Evans, R., et al.: Making sense of raw input. Artif. Intell. **299** (2021)
24. Bennett, M.T.: Compression, the fermi paradox and artificial super-intelligence. In: Goertzel, B., Iklé, M., Potapov, A. (eds.) AGI 2021. LNCS (LNAI), vol. 13154, pp. 41–44. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-93758-4_5
25. Delétang, G., et al.: Neural Networks and the Chomsky Hierarchy (2022)
26. Power, A., et al.: Grokking: generalization beyond overfitting on small algorithmic datasets. In: ICLR (2022)