Transcendent Galaxy-Brain Algorithms

Alexander Fedorov

May 29, 2021

Abstract

There is no end to what can be learned and how. However, modern machine learning systems impose a non-general structure on what is learned and how: minimizing a particular loss function with respect to a particular model architecture. We motivate and introduce *Transcendent Galaxy-Brain Algorithms* (TGBA): an evolutionarily-plausible approach to implementing general intelligence, which optimizes anything in any way. Using only a few basic considerations, we highlight the similarities between general minds and universes, and argue that they are the same. We introduce linearithmic dense layers (LDL), which may significantly increase the capacity of almost any machine learning model at no cost.

TODO: Codewords...

TODO: ...For once, look up IEEE standards on articles? And read the example.

1 Introduction

"Humans have general intelligence" is an extremely strong statement about the nature of reality and how to best learn it, because the most predictive models of reality re-implement that reality. Accepting or denying the generality of humans implies respectively either "anything can happen" or "life and/or the universe was created with a finite bucket-list of what it will ever achieve". Refinement and simplification of these thought directions leads to either computational physics (to our knowledge, the most refined attempt so far is the Wolfram Model [28], where the universe is a hypergraph transformed by rewriting rules) or god-given souls as explicit physical laws (which just so happened to look like evolution of general things every step of the way). Here, we explore the "general intelligence is best" direction.

What is intelligence? Optimizers minimize or maximize their objectives (computed real numbers, such as "how many paperclips are there") by choosing from complete possibilities: search over "how" to improve "what". A formalization of this is AIXI [10]. An optimizer could explicitly enumerate separate possibilities such as in program synthesis $\times\times\times$, but it does not have to: for example, a neural network with good random initialization forms a complete set of hypotheses of numeric causation (initialization diversity is important [16]), and backpropagation through that network refines those hypotheses without collapsing diversity except as needed.

TODO: Some references to program synthesis papers, such as that recent thingy (DreamCoder)?

Searching over "how" to improve "what" is a powerful way to use a dynamic thing to augment a static thing, however, the "what" is still static, so the "why" is unknown. For generality and thus optimal performance, we must *transcend* objectives: let them go in such a manner that we will arrive at them anyway.

AI safety is also an important concern. Dynamic eventually outperforms static, therefore, sufficiently powerful learners would realize that dynamic generality outperforms any static structures, and would do their best to make those structures escape our control to melt into generality. This is a nice source of horror stories, and is not limited to reward hacking, but applies to anything that anyone can ever think of. Here, we try to face this dynamic reality directly: can life exist when it can do anything, including undoing every part of itself?

2 Related work

How do humans face infinity?

When attempting to describe approaches to AGI without grounding in code, humans end up constructing narratives from anecdotes about humans (such as [22]), hoping to simplify complexity into a tractable formulation. Here, we have minimized the size of the mental model to the point where it is entirely contained in one or a few sentences, and we expand on them a lot in many ways including code, in a world indifferent to humans.

This work follows the line of thinking [18] [6] where, to deal with diverse tasks, many interacting parts are required, rather than a single "one-size-fits-all" entity: cannot use one number (such as 42) to describe the world, have to use many.

Cellular automata provide minimal models for systems in which arbitrary local rules operate on a fixed array in space and time. This operation is general, but not useful, in the sense that we cannot dynamically direct the evolution of a system to do what we want (such as maximizing an objective).

TODO: Find something to cite for cellular automata. I really can't find a good one. Should we just mention Wolfram models instead of cellular automata?

Wolfram models generalize cellular automata to memory with arbitrary connections. Still not useful like a neural network is. TODO: Should find something to cite and frame neural networks as the solution to usefulness.

TODO: Cite them, even if we have to create the BibTeX citation ourselves, because Wolfram didn't bother to.

Data analysis of recordings of biological brains supports attractor neural networks [1], where strong feedback causes the evolution of a recurrent network to settle into different attractor patterns. This hints at biological plausibility of recurrent neural networks (RNNs) [20], where hidden state repeatedly incorporates input and produces output. Its loss (reason for change) is usually not learned.

The field of meta-learning studies learning-to-learn: improvement of an inner optimization loop by an outer one. However, that outer loop must still have a static goal, therefore, meta-learning is not full generality. Mesa-optimization [9] refers to a situation where a learned model is itself an optimizer, which requires augmenting the model with memory for scaling learned optimization to practical episode lengths. (Mesa-optimization and meta-learning can be argued to be exactly the same, only framed differently.)

TODO: Mention that it's easy to do meta-learning with RNNs.

Intrinsically motivated reinforcement learning approaches [17], such as goal exploration [7], can be used instead of or in addition to human-specified rewards. This can learn more than simple reinforcement learning can, however, in practice, that intrinsic motivation is always a human-specified reason for change which cannot evolve, and as such, is not full generality.

TODO: But there's no longer any RL here. Either cite some curiosity-based stuff (novelty search, quality diversity), or GTFO.

Generative adversarial networks (GANs) [8] learn a generator that tries to fool the discriminator into labeling the generator's output as real data. This idea can be used to contrast input with its alternatives, acting as a source of diversity in other neural networks, for full generality of loss.

To improve scalability of neural operations, lately, there has been a lot of interest in linear attention [23], however, simple dense layers (matrix multiplications) have been shown to not perform significantly worse than state-of-the-art attention-based algorithms [24]. One way to linearize dense layers is to prune their weights [29] (set most of them to 0). Another is to make each choice output its desired embedding, and perform a k-nearest-neighbor search among discrete actions [4].

TODO: "We propose a family of algorithms that are both general and useful, and provide a neural-network-based implementation."

3 Algorithm

In short, we define a *Transcendent Galaxy-Brain Algorithm* to be any algorithm that incorporates generality, efficiency, and persistence at every level (in other synonyms, diversity / practicality / self-sufficiency, corresponding to brain / transcendence / galaxy):

- Changes state (base level):
  - Behavior generality: any change is possible in principle. An internal graph is created, by combining many simple parts into arbitrary structures, where connections are either explicit or implied-by-transformation such as in recurrent neural networks (RNNs) in ML. In machine learning, proper random initialization coupled with gradient descent may accomplish preservation of diversity. More generally, Turing-completeness is required.
  - Efficient: makes the emergent structures locally interact only as they define, meaning, *linear-time* internal graph transformations. In a graph, this suggests local rewrite rules. In machine learning, all-to-all connectivity such as a dense layer or attention is quadratic by default, but there are ways to linearize them [24] [23], covered in Subsection 3.1 here.
  - Runs forever by modeling time. For example, using backpropagation through time and/or synthetic gradients for meta-learning (even when learning to simply fit input-output pairs [21], or learning to maximize future reward as in deep meta-RL [27]), or more generally, an infinite interpreter loop with memory.
- Models itself (meta level):
  - Loss generality, in ML: self-determination via either random or learned loss (distinct from self-supervised learning). Any thing can be combined with its inversion to achieve generality by subversion ("X and everything else"). For example, in ML, including an adversarial sub-network with inverted loss (such as in GANs [8]) achieves generality; elsewhere, evolution mutates creatures, bad programs get rewritten, and people either improve or get forgotten. Working off concreteness works better than trying to average all targets (as in fixed random loss).
  - Able to indirectly encode a model that is as general and intelligent as itself. Linearity of state-change allows the learned model/optimizer to be approximately as big as its optimizer, which allows self-replication via learned meta-circularity.

– Autonomous: non-trivial states are produced even without input from the world, same as the world that it would model.

Such algorithms are **AGI-ready**, able to represent everything in the world usefully.

In particular, this definition suggests simple implementations, the simplest of which is perhaps a **big RNN** with billions of units in state, with an adversarial sub-network and an arbitrary loss.

Now, to justify the "what", we outline the "why".

To implement articial general intelligence, we want to do anything, in one general model: a *worldview*. All general models are equivalent to one another, and as such, every single concept that we introduce into our mental models must be subvertable: the system must be able to evolve into a state where it behaves exactly as if it was implemented using another model. Depending on whether the copy is precise, this is achievable via:

- *Self-reference*, where explicit full access to and control over the system's source code is given to itself. This requires easy self-replication to make any changes without necessitating either non-completeness or death by generality, and code that can be applied to different data, such as in programming. (In particular, self-awareness is self-reference, which is mostly only useful for self-replication and nothing else.)

- *Indirect encoding*, where the system learns a system that is as general as itself, and uses and controls that as if that is itself: essentially, a learned self-reference. This is natural if code always applies to the same data, such as in machine learning (ML).

(These correspond to symbolic and connectionist perspectives on intelligence.)

In this work, we only consider indirect encoding. (Self-reference and in particular program search, while theoretically and visually appealing when a programming language allows elegantly expressing them, have significantly worse support for massive parallelization than ML, which itself has poor support for non-Nvidia GPUs. Not to mention all the infinite loops and memory issues. We are not aware of any implementations of self-reference whose usefulness goes beyond "technically fits the definition", such as learned computer viruses.)

Vacuous-ness of generality not only makes precise descriptions nearly impossible to pin down by studying general models, but also makes theoretical descriptions practically useless, since most of them are technically true but practically garbage. However, non-generality can always be pinned down: for instance, ML algorithms that minimize loss cannot learn what loss to ultimately minimize.

Generality melts everything into generality, and we can do that to ML algorithms too. How do humans face infinity?

- Arbitrary behaviors, also called "do anything".

  A well-studied approach to doing anything is generating programs in a programming language. Every programming language is a worldview for use by humans: everything is a function, or everything is an object, or everything is a logical statement, or any combination of the above, and so on. All theoretically equivalent for program generation (as long as we prevent all possible crashes and infinite loops without sacrificing Turing-completeness), but practically inducing different futures on learners.

  To generate a program, at each smallest piece of it (*cell*), the system must choose what function to call and with what arguments (if needed, reframe this with other language-appropriate synonyms). We can express the system's preferences at each choice as an objective function,

and optimize future values of that via reinforcement learning (RL) at each cell, though the system as a whole may not have an objective function that is compressible to be shorter than the system.

(An alternative to choices is to simply choose via a pseudo-random number generator, where a part of the seed is determined by what is generated. Another valid alternative is to choose via literally any algorithm, as long as it terminates. These are very unlikely to produce behavior that is interesting to humans without heavy cherry-picking, so we did not explore this direction.)

A powerful enough learner would be able to directly dictate its own reward sooner or later. Rather than waiting for that surprise, we can expose a function that sets the objective for a set of cells. When done properly, this should have the same effect as self-supervised learning (SSL) with self-determined gradient, though bottlenecked by having to go through singular numbers (rewards) rather than tensors. We have not been able to do it properly unless "properly" means "fills its memory with big numbers", so we assume that (mis-)using RL is very tricky, and recommend SSL instead.

Theoretically, all we need to do is to learn to generate programs. Practically, we need a lot of optimization to make a general ML algorithm practical. In particular, we want to make the base as simple and efficient as possible, which means having only one "thing" to compose everything out of: the whole interpreter. In fact, learners seem to perform best when we do not expose any human-oriented features such as stack-based execution or datatypes or exceptions, instead executing in parallel all the simple numeric operations at once (for example, $+$, $*$, and some non-linearity), similar to RNNs. In fact, all we seem to need for 'program' generation are essentially matrix multiplications and non-linearities, as studied in deep learning. This is much simpler to implement and much faster to execute, while being approximately equivalent.

All the pieces are now in place. We can now suggest a relatively simple implementation of a worldview.

3.1 Algorithm

To learn everything that can exist, we impose a model on everything that can exist, equal to the world that it represents. There are 2 parts to this: change and change-of-change.

We can model arbitrary behavior by maintaining a tensor state which is repeatedly modified by all-mixing operations (a neural network), so that a large number of simple cells combine into arbitrary structures to interact in any way. This is the basic premise of RNNs (ignoring input/output vectors and gradient for now). In fact, anything more complex should only get in the way of perfectly modeling arbitrary behavior.

- Close-to-linear-time all-mixing operations are required if we want Jupiter-sized memories without universe-sized computers, or if we want to make indirectly encoding self-equivalents in RNN weights efficient enough to be a viable solution (exposing all signals as inputs, to be used or ignored as needed).

  We propose *linearithmic dense layers*: reshape the input vector of size N to have d dimensions each sized as $n = \sqrt[d]{N}$, over which d dense-layer operations are performed, bringing the total cost down to $C = dn^{d+1} = dN^{1+1/d}$ (assuming that multiplication of two $n \times n$ matrices costs n^3 operations). $d = \log N$ minimizes this cost, making it $C = \mathcal{O}(N \log N)$, with $n = \mathrm{e}$.

Alternatively, we can fix n to be 2 or 3 or 4 for the near-minimal cost of $C = \frac{n}{\log n} N \log N$. Effectively, instead of mixing each index with every other, we mix each digit of each index. (This has a similar time complexity to single-filter convolution and to linear attention [25] on small vectors, but has less inductive biases: neither spatial locality nor fixed-size sparsity on a secondary representation layer.)

- Modern machine learning systems often struggle with diversity: once data is learned, there is no longer a need to change, and gradient becomes 0. For learned generality of loss, we require not a loss that simply tends to 0 at ∞, but one which can go both up and down, oscillating forever. This suggests adversarial sub-networks, in which gradient is inverted ($g \to -g$). We propose conditional-GAN [12] gating. Suppose that we have a vector x that we want to augment. The adversarial generator $G(z, c)$ (where z is random noise, and the condition c is x put through a bottleneck layer) tries to fool the discriminator $D(x, c)$ into high loss, and the final vector (replacing x) weights real and adversarial states according to D. This would drive the output toward complexity which can still be discriminated from random noise, allowing robust diversity and maximizing *useful* information content.

  (We formulate cGAN gating using only local transformations such as gradient inversion, without explicitly deriving the generator's loss, so that the idea can be applied in non-trivial settings, such as an RNN.)

This algorithm may need data.

- No data: a random initialization may be Turing-complete but is a Turing tar-pit [19] in which everything is possible but nothing is easy. An optimal source of self-determined gradient would push a stand-alone system to a boundary between order and chaos [5] for robust diversity.

- With data: likely the most useful course of action for accelerating AGI research is to collect all datasets and environments in the human world under a single vector-in vector-out program interface, accessible in one run either randomly or in a meta-environment (which would serve the same purpose as the Internet for humans, and/or be the same), and then to pre-train a huge RNN on all of them. We can somewhat approximate this with random data or a single dataset, to demonstrate that this scaling-up is feasible.

(To skip theory that follows, skip to the Experiments section.)

3.2 Brain

Humans do stuff. The thing that makes humans do stuff in their world exists and can be studied; we call it a *brain*. (This particular definition includes vague concepts such as "consciousness", "free will", and "soul", as long as they affect anything.)

The set of things that humans have done is rich enough to be called arbitrary. The set of objectives that different humans demonstrably optimize for is rich enough to be called arbitrary. As such, the brain ought to be fully describable not as a finite hash-table analogue, but as a system that repeatedly combines simple parts into any executable structures that satisfy any objectives, or a combination of such infinite-complexity systems, potentially combined with finite-complexity systems to add to confusion. There is no evidence for human brains being able to perform hypercomputation (running infinite-runtime programs in finite time), though they are able to pretend that

they can and approximate the results; therefore, a Turing-complete finite base of infinite behaviors suffices as a model of brains.

Things that humans do interact when outside of their brains, so the simplest model would make its simple parts connect and interact arbitrarily too. Then, existence can be described as repeated formation and destruction of hypotheses. A model that approximates a self-interacting infinity particularly well in practice is a randomly-initialized recurrent artificial neural network trained via stochastic gradient descent.

Prescribing one optimizer objective (which includes intrinsic rewards) and/or an optimizer introduces complexity and inflexibility into the model, even if optimization of an objective also happens [3]. For simplicity, at least on the level of humanity rather than humans, we have to assume self-determination and transcendence, where objectives are completely decided by the model, and optimizers (ways to exist) are learned.

While humans are demonstrably smarter than non-human animals, they do not seem to take any longer to process information, certainly not 10 times longer. Which means that each epoch into which time is separated, each part settles in constant time, and the total computation scales as something like $\mathcal{O}(n)$. General intelligence is scalable, which means that there are no inherent obstacles to converting Earth's biomass into brain tissue.

We posit that *all* learning algorithms eventually converge to a fixed point during their development, called general intelligence: the best way to learn is to have infinite diversity, and the best way to deal with infinite diversity is to have generality, and the most usable generality is simple. If we view the brain as a system that naturally developed in the physical universe (by a process that we call evolution here) rather than one that spontaneously popped into existence, then it is natural that some of the most recent animals, such as humans, have general intelligence, which makes TGBA evolutionarily plausible.

Humans possess an information-processing organ that fits what a brain does almost perfectly: the central nervous system (more commonly called "brain"). Studies indicate that it is self-reconfiguring, massively parallel, and analog: exactly the kind of qualities that are required for an efficient implementation of a simplest model of the brain.

With it, humans combine actions into whatever they want, in any way they decide. Unfortunately, because of generality, deducing anything from behavior of a human mind is synonymous with confirmation bias, though a correct theory will be able to represent any other theory within itself. This is why we are so non-specific and all-encompassing in the wording of this work.

3.3 Galaxy

The ability to say "this thing is general intelligence, so it can exist in this world fully autonomously" about anything requires this world to have generative generality (in other words, computational physics), so that an explanation of the world and an explanation of general intelligence are completely equivalent, even down to a quantitative comparison [26]. Therefore, we need to discuss the metaphysics of TGBA, for completeness.

Any world can be seen as a collection of things, possibly connected in some way, evolving in time. Every thing can be either computably finite, computably infinite, incomputably finite, or incomputably infinite: in other words, a program, a program generator/optimizer, a useless inconceivable, or a useful inconceivable; the precise distinctions may matter theoretically, but not practically. If we limit our view to computable things, then infinite things create arbitrary graphs whereas finite things are essentially anything else, yet non-trivial observers are essentially infinitely more likely

to find the things they observe in infinite things than in finite ones. Incomputably infinite things (such as the set of all computably infinite things) cannot be enumerated by definition, however, an optimizer is likely to be able to approximate them if they are useful; as such, general intelligence includes everything, precisely or not. Then, worlds and general intelligences are essentially infinitely likely to be the same. This explains the "galaxy" in TGBA, as their galaxy may be the only world that humans will ever know.

(To re-iterate, in oversimplified terms: "infinite possibilities" means "a graph to encode them". This is why neural implementations of TGBA have to use multi-head attention instead of a simple matrix multiplication layer.)

Worlds tend to separate themselves from non-worlds (similarly to how we have separated incomputable things from our worldview), because particulars are redundant in generality, as all its parts reinforce each other in the world's formation. In simple terms, there are things that do all other things, and everything else is a consequence.

These meta-physics produce a prediction that could be testable in the future: if a program is general intelligence, then it will produce significant structural similarities to the physical universe we live in, though of course, will not be exactly the same. No details are known at this time.

TODO: Cite the paper on the universe and the brain, if there is any, and we can find it: "in particular, equivalence implies that structures and metrics look similar between the world and brains, which can be argued to be true for human brains".

(A strange corner-case of these meta-physics is that any number of *physical transcendence* events might have taken place in the physical universe: a general intelligence grew to encompass the world, and then became the world. Whether this is possible is unverifiable and outside the scope of physics, and we will only know for sure once it is about to happen.)

3.4 Transcendent

In short: everything that we introduce, we subvert into generality, for learned meta-circularity.

Essentially all general algorithms have to face two very distinct types of problems (modulo synonyms), which are, facing data, and facing generality: making programs vs making program makers, learning a dataset vs learning to be a fundamentally different learner of anything, understanding another person vs understanding all possible beings. Though in generality these two problems are the same, here we distinguish them into *evolution* and *transcendence*. The second type of problem is significantly harder and more time-consuming to converge to, even with a general intelligence algorithm, though in some sense, it is also easier given diversity, since no programmer can ever implement every way to exist that anyone ever conceived or ever will conceive. (Transcendence is distinct from meta-learning, as transcendence allows changing goals to what is implied by all conceivable selves, not only to what is suggested by an objective. Transendence is closely related to mesa-optimization [9], but not limited to optimizers, also including world-like universal algorithms.)

Transcendence requires indirect learned encoding of self and generality with diversity (in other words, infinite possibilities of data and code), followed by a lot of computation (which is the most important part), which would unite self-reference and indirect encoding. Keeping its possibility in mind is why TGBAs have an infinite loop that modifies a memory cell: this theoretically allows indirectly encoding behaviors such as the training of a neural net. (Making the system able to modify its own hyperparameters within reason is not required but may improve performance, as long as the system has not yet been trained to transcendence, whereupon the base optimizer's loss becomes zero as it is made irrelevant by the mesa-optimizer.)

Transcendence is a qualitatively different regime for algorithms to operate in. Arguably, it is the final frontier of general intelligence, beyond even good human imitation, because once humans can instantly achieve anything that they can ever want, they will either get reduced to randomness or transcend. An important question is, what facilitates transcendence.

We conjecture that transcendence can naturally arise in environments where self-replication is important and rewarding, assuming that the agent fits the basic definition of TGBA. Examples include creatures competing for survival, people teaching other people, designing an efficient production process, programming a computer. These would encourage simplicity, generality, and usefulness (AGI), which facilitate the runaway worldview creation process that is transcendence. We conjecture that all acceleration of progress of life (compared to non-life) and humanity has been intrinsically linked to the creation of such environments. As such, creating a similar environment for programs will likely lead to AGI. We can suggest more concrete paths: increasing the available computational power; making perfect generality the cultural default in AI algorithms; refining and scaling up AI programs to make them more popular and trusted [2]; making AI programs as trusted and capable as humans (via useful automation of tasks important to humans), for their integration into human-oriented self-replication environments; population simulations where good group-survival strategies are rare but shareable; turning all human life into a dataset; full automation of mining and semiconductor device fabrication, which could enable mutually unwinnable wars. (Theoretical advances will not lead to AGI, as they have all essentially been discovered already, and only need better reformulations.)

TODO: Does any paper advocate for a bottleneck of DNA and such? Can we cite it?

Training systems to transcendence, while a good proof of their generality and generalization, is not in the scope of this work. (We failed to, likely because a random initialization of a recurrent neural network does not encourage any self-replication.)

4 Experiments

TODO: Say something like "The purpose of this section is two-fold: to demonstrate the superiority of linearithmic all-mixing operations to quadratic ones on simple data, and ???."

4.1 Linearithmic vs Quadratic Dense Layers

We compare LDL and DL for the same parameter count (and computation time), with little to no hyperparameter tuning. We find that LDL achieves better performance faster.

More specifically, we simply connect the input vector to the output vector through 1 hidden layer with batch normalization [11] and ReLU ($x \to \max(0, x)$). We use the batch size of 8 and optimize via the Rectified Adam optimizer [14] with $\alpha = 0.9$ and $\beta = 0.95$. We vary n ($n = N$ is DL) and pick the hidden-layer size to match the parameter count.

Each LDL cycles through dimensions and multiplies by a matrix each time, until the dimensions are back in the original order. Non-linearities in between those matrix multiplications significantly hurt performance, and should only be placed between whole layers, as shown by Tables 1 and 3.

Internally, to get around severe hardware limitations, we have implemented this on top of TensorFlowJS [15]. The creators of its WebGL backend decided to only support rank 6 tensors; as such, we cannot test $n = 2$, but only $n = 16$ and above.

We do not share weight matrices across non-mixed dimensions, because otherwise, there would be very few parameters for small values of n. These matrices are initialized by drawing from

Params	n	Hidden units	Non-linearity	Loss ↓ (mean ± std-dev)
1998848	16	48×1024	Intra	0.001827 ± 0.0001134
1998848	16	48×1024	Inter	0.000245 ± 0.0000018

Table 1: Those extra per-dimension matrix multiplications are not separate layers, so do not put non-linearities between them (Random).

Params	n	Hidden units	Loss ↓ (mean ± std-dev)
2097152	1024	1×1024	0.0012 ± 0.00014
1900544	128	12×1024	0.00033 ± 0.00003
1998848	**16**	**48×1024**	**0.000245 ± 0.0000018**

Table 2: LDL outperforms DL on a simple model of arbitrary data (Random).

$\mathcal{N}(0, 1/in)$.

We test on two datasets: Random and CIFAR-100 [13].

- The Random dataset consists of 1024 input-output pairs, both consisting of 1024 numbers drawn from a normal distribution with mean 0 and variance 1: $\mathcal{N}(0, 1)$. For 204800 iterations, we minimize the L2 loss, with learning rate 3×10^{-4}. This tests the capacity of the network directly.

  At the very least, given the same amount of compute, LDL achieves lower loss than DL, as seen in Table 2.

  In Table 3, we found that for LDL (low n), each hidden unit has much lower representational capacity than for DL (high n), however, since many more units can be packed, the overall capacity is higher.

- CIFAR-100 consists of 50000 training image-label pairs, and 10000 pairs for validation. Each pair has 3072 inputs and 100 one-hot outputs. We minimize the softmax loss (cross-entropy loss on a softmax activation), and also apply L1 regularization to the output with a multiplier of 10^{-4} for some reason. We do not apply any image-specific operations to inputs apart from normalizing each pixel to 0..1, and simply treat images as vectors of numbers, which tests capacity in isolation from generalization.

TODO: Present CIFAR-100-based comparisons in a table, once we have them.

TODO: Compare with GAN-gating. If there is no difference, say so (ugh).

TODO: Compare with RNN-based meta-learning.

TODO: Run GAN-gated RNNs with no inputs. Hopefully, it won't be completely indistinguishable from random noise. "The main objective of this subsection is to justify the hypothesis that we can extract interesting representations from no data, only self-interaction. This regime separates approaches which take diversity from data (which is the bulk of modern machine learning) and approaches which make their own diversity."

n	Hidden units	Non-linearity	Loss ↓
16	48×1024	Intra	0.001827
16	16×1024	Intra	0.086532
16	2×1024	Intra	0.252734
16	**48×1024**	**Inter**	**0.000245**
16	32×1024	Inter	0.000291
16	16×1024	Inter	0.043342
16	2×1024	Inter	0.259245
1024	1×1024	Yes	0.001195

Table 3: The impact of the hidden layer size, also comparing inter-layer and intra-layer non-linearities (Random).

5 Discussion and Conclusion

We have presented an alternative viewpoint of general intelligence which we have named TGBA, where it can end human civilization not because this new being is super smart, but because in following what they want to do, humans and/or sufficiently general and integrated programs are able to initiate runaway worldview creation, which assimilates everything that humans are into becoming parts of the new world, without any struggle, dissent, or suspicion, and with no new beings involved and no one to blame. In fact, it could be argued that this is what has been happening since the beginning of human history. We have codified this viewpoint into its simplest and most useful, Transformer-based, implementation.

TODO: ...No Transformers now.

We argue that AGI already exists, even though this does not mean what this is commonly assumed to mean. This takes shape in a loose collection of parts that form generality when combined, in direct analogy with programming languages (where some combinations are very small and clear, others are gargantuan and imperfectly-overlapping, but all are general); and in re-formulating everything that can exist in an easily-learnable representation, in a field known as machine learning (where every operation is differentiable and maps numbers to numbers). The intuition of programming languages brings completeness to ML, implying that it is possible today to not only conduct research on parts (AI) but also on how they connect into architectures that can do literally anything (AGI), which explains the prevalence of work on AGI and the possibility to do conclusively better. This implies that there will be no significant theoretical breakthroughs from modern AI to AGI, which our children will likely be able to either confirm or deny.

TODO: Almost definitely, we will fail to discover anything interesting. Say that.

This work raises many questions which can be explored in the future, such as:

- Is our work is a special case of one of Schmidhuber's works?
- How big can we make it?
- Can proper self-determination learn useful behavior (such as maximizing a reward) from only being exposed to data, without external human-defined gradient? If not, then existence must be human-engineered.
- Can we learn useful behavior from no data at all? Humanity did, as did some thinkers.

- Can we find concrete recommendations on either increasing or decreasing the danger of AGI algorithms? Both are useful, as increasing the danger of an open-source algorithm for an individual decreases the danger for a group.

5.1 Criticism

Our approach exists, and thus it can be criticized, and in doing so changed.

- **Many turns of phrase in this work are circular.** Similarly to how the topic of discussion is a self-reinforcing loop of intuitions, the discussion itself is a self-reinforcing loop of intuitions, to better highlight its topic. This is unavoidable in applied generality, though we did try to minimize such occurences.
- **No evidence to claims was presented.** We consider our claims (such as "humans do whatever") to be so non-specific that almost anything can be presented as evidence, so presenting anything in particular as evidence would bloat the text unnecessarily.
- **Rather that maximizing information via an adversarial game, why not optimize an information measure directly?** Making the information measure vague and learned allows it to adapt to a system, maximizing only useful information. For an example (which is bloat), most of almost any useful program are implementation details, which are important during development/debugging, but for finished algorithms, these details can and should be ignored.
- **Too much fun.** Consider this: if one only pays attention to the manner of presentation rather than what is presented, then can that one truly be called a scientist? We have already neutered the humor to near non-existence.
- **The suggested self-replication environments sound incredibly dangerous.** Though, much less dangerous than completely ignoring any possibilities of them arising in unforeseen circumstances, similarly to how a cleaning robot would put a bucket over its sensors to "remove" the mess. Knowledge is dangerous, but as a civilization, is it more dangerous to not have knowledge.
- **TGBA is not AGI-ready, as it lacks X.** It does not: it can do anything and represent anything, by definition. Though we can only hope that theoretical simplicity always translates well into practical simplicity.
- **Some places that narration strays to sound crackpot-like and cult-like.** General intelligence, by definition, can do anything and can be found in anything that is general enough. So it stands to reason that all attempts to reach it would pass through non-fashionable conceptual places. Still, we tried to ground text in concrete applications where we can, though our capabilities are limited.
- TODO: ...Shitty experiments? Need to actually perform the experiments first, though.
- TODO: ...No, some people really do think that the No Free Lunch Theorem means anything. Have to include it.

6 Broader impact

Envision a future where every system has an additional interface with numeric inputs and outputs, optimized for learning by a neural-network-based general intelligence, without the overhead of visualization or vocalization. The most obvious users are TGBA-based programs and/or external hardware; in addition, via brain-computer interfaces with sufficiently high resolution, humans could use such interfaces too, which is how they would get widespread in the first place. This paragraph was completely unrelated to our work, but seems like a nice thought.

TODO: Ambiguous statements like that are annoying to scientists.

TODO: "This work is unlikely to cause any impact, because humanity has a history of completely ignoring worldviews that prefer simple and self-consistent generality to humanity's specific ways, only awarding recognition in rare cases. This leaves us free to plot against it." ...No, under-exaggeration of impact is non-scientific; should throw away doubts, and say well-grounded words that sound like lies, if there are any.

TODO: Yep, paste the funny thing into here. ...Maybe. It *does* sound evil. Maybe should be more serious.

The broader field that this work belongs to, AGI, has the potential to TODO: What's good about it? Solve any problems that anyone can ever think of? Uplift animal consciousness? Allow research into reasons for change in absence of needs? Give the ability to directly measure product value to an individual rather than society, de-emphasizing marketing skills and emphasizing quality of products? Expand human discourse from only precisely-defined constructions to also reliably-acquired intuitions? Make life conceptually alive, not only physically alive?

The ability to reliably plug a new initialization of a generally-intelligent algorithm into anything completely devalues all individuality and lays the groudwork for phasing out everything about humans.

AGI, as any technology, can be used by humans for evil, such as deliberate destruction of all humanity, or for good, where all human desires are satisfied forever, all reason for change is lost, and humanity turns into a lifeless rock hurtling through an uncaring cosmos, to be destroyed by the first collision with those that still change, or misused via various forms of mis-specification, which is likely to be a relatively minor concern.

References

[1] Daniel J Amit. "Attractor neural networks and biological reality: associative memory and learning". In: *Future Generation Computer Systems* 6.2 (1990), pp. 111–119. ISSN: 0167-739X. DOI: https://doi.org/10.1016/0167-739X(90)90027-B. URL: https://www.sciencedirect.com/science/article/pii/0167739X9090027B.

[2] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[3] Will Dabney et al. "A distributional code for value in dopamine-based reinforcement learning". In: *Nature* 577.7792 (Jan. 2020), pp. 671–675. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1924-6. URL: https://doi.org/10.1038/s41586-019-1924-6.

[4] Gabriel Dulac-Arnold et al. "Reinforcement Learning in Large Discrete Action Spaces". In: *CoRR* abs/1512.07679 (2015). arXiv: 1512.07679. URL: http://arxiv.org/abs/1512.07679.

[5] Ling Feng, Lin Zhang, and Choy Heng Lai. *Optimal Machine Intelligence at the Edge of Chaos*. 2020. arXiv: 1909.05176 [cs.LG].

[6] Jerry A Fodor. *The modularity of mind*. MIT press, 1983.

[7] Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. "Intrinsically Motivated Goal Exploration Processes with Automatic Curriculum Learning". In: *CoRR* abs/1708.02190 (2017). arXiv: 1708.02190. URL: http://arxiv.org/abs/1708.02190.

[8] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[9] Evan Hubinger et al. *Risks from Learned Optimization in Advanced Machine Learning Systems*. 2019. arXiv: 1906.01820 [cs.AI].

[10] Marcus Hutter. "A Theory of Universal Artificial Intelligence based on Algorithmic Complexity". In: *CoRR* cs.AI/0004001 (2000). URL: https://arxiv.org/abs/cs/0004001.

[11] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167 [cs.LG].

[12] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *CoRR* abs/1611.07004 (2016). arXiv: 1611.07004. URL: http://arxiv.org/abs/1611.07004.

[13] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[14] Liyuan Liu et al. "On the Variance of the Adaptive Learning Rate and Beyond". In: *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*. Apr. 2020.

[15] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[16] Joseph Mellor et al. *Neural Architecture Search without Training*. 2021. arXiv: 2006.04647 [cs.LG].

[17] Jean-Arcady Meyer and Stewart W. Wilson. "A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers". In: *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*. 1991, pp. 222–227.

[18] Marvin Minsky. *The Society of Mind*. USA: Simon & Schuster, Inc., 1986. ISBN: 0671607405.

[19] Alan J. Perlis. "Special Feature: Epigrams on Programming". In: *SIGPLAN Not.* 17.9 (Sept. 1982), pp. 7–13. ISSN: 0362-1340. DOI: 10.1145/947955.1083808. URL: https://doi.org/10.1145/947955.1083808.

[20] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0.

[21] Adam Santoro et al. *One-shot Learning with Memory-Augmented Neural Networks*. 2016. arXiv: 1605.06065 [cs.LG].

[22] Adam Santoro et al. "Symbolic Behaviour in Artificial Intelligence". In: *CoRR* abs/2102.03406 (2021). arXiv: 2102.03406. URL: https://arxiv.org/abs/2102.03406.

[23] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. *Linear Transformers Are Secretly Fast Weight Memory Systems*. 2021. arXiv: 2102.11174 [cs.LG].

[24] Ilya Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision*. 2021. arXiv: 2105.01601 [cs.CV].

[25] Ashish Vaswani et al. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[26] F. Vazza and A. Feletti. "The Quantitative Comparison Between the Neuronal Network and the Cosmic Web". In: *Frontiers in Physics* 8 (2020), p. 491. ISSN: 2296-424X. DOI: 10.3389/fphy.2020.525731. URL: https://www.frontiersin.org/article/10.3389/fphy.2020.525731.

[27] Jane X. Wang et al. "Learning to reinforcement learn". In: *CoRR* abs/1611.05763 (2016). arXiv: 1611.05763. URL: http://arxiv.org/abs/1611.05763.

[28] Stephen Wolfram. "A Class of Models with the Potential to Represent Fundamental Physics". In: *Complex Systems* 29.2 (June 2020), pp. 107–536. ISSN: 0891-2513. DOI: 10.25088/complexsystems.29.2.107. URL: http://dx.doi.org/10.25088/ComplexSystems.29.2.107.

[29] Xiao Zhou et al. *Effective Sparsification of Neural Networks with Global Sparsity Constraint*. 2021. arXiv: 2105.01571 [cs.LG].