Learning World Models

Linas Vepstas

19 October 2023

Hi Greg,

This PDF picks up where the plain-ASCII email leaves off.

Without further ado: you're interested in formalizing "learning". This will require drawing careful distinctions between related concepts. So, here we go. My apologies, this is a long text and it took me six hours to write it. Ooops.

Starting point is the "good regulator theorem": https://en.wikipedia.org/wiki/Good_regulator Cut-n-paste: "every good regulator must contain a model of the system", where by "system" it is meant any one of several things:

- The "system" is the "external world".
- The "system" is the "input-agent-output" triple.
- The "model of the system" is a theory of the "external world" which captures "when action X is performed, Y will happen in the external world" (observe that I ignore "input" in this third definition: the model only needs to capture how the world reacts to actions; a model of the input mechanism is not needed, at least, not to first order.)

There is much to be confused about here. So I will attempt to formalize.

Definitions

The following section (and also the following six-seven sections) attempt to formalize and provide a notation for conventional concepts in the AI and ML world. I've attempted to stick to textbook concepts, and to avoid circular definitions, but perhaps I've failed. This is a sketch. Later sections leave the beaten track.

- W ::= external world, which is inaccessible except through sensors. (Example: a temperature sensor provides temperature data about this world, but, like all sensors, the observation can never be omniscient.)
- B ::= a database ("inside" the agent) that holds state information about the external world. (Example: it is capable of holding a single integer, which will be the most recently sensed temperature. It could be a time-sequence of temperatures. It could be a collection of Bayesian priors about what the external temperature

might be.) Standard AI texts give this the name of "working memory". It is a scratch-pad, a place where important time-varying results are kept. In addition to this, there are other "places" where "knowledge" is "kept", to be elucidated later.

- H ::= a collection of hypotheses about the external world. For example, h_1 is the hypothesis "the temperature is 63 degrees F" and h_2 is the hypothesis "the temperature is 67 degrees F". A direct sum symbol $\oplus$ is used to enumerate the elements in the collection, so $H = h_1 \oplus h_2 \oplus \cdots$. The direct sum symbol is used because the hypotheses are intended to be mutually exclusive: it can be 63 degrees or 67 degrees but not both. It is fun to call H the space of "many worlds", precisely because these worlds are mutually exclusive. It is intellectually risky to call H a "set", and it is misleading to write $H = \{h_1, h_2, \cdots\}$. Doing so risks miscommunication and confusion. However, I will sometimes call H "a set", because I want to use membership: $h \in H$ is a "possible world".

- A "Bayesian prior" $\mu : H \to \mathbb{R}$ that assigns a real number to each $h \in H$. Notation: $\mu : h \mapsto x$ where $x = \mu(h)$. I use the symbol μ because it is suggestive of "measure". Now, "measure" is "the same thing" as "probability", but if one uses the symbol p for every freakin probability that shows up, the proliferation of p's is confusing. I will reserve p for posteriors or other probabilities, and reserve μ for priors. I strongly wish μ to be interpreted as a measure upon some sigma algebra. So, for example: $\mu(h_1 \cup h_2) = \mu(h_1) + \mu(h_2)$ and $\mu(h_1 \cap h_2) = 0$ because $h_1 \cap h_2 = \varnothing$ because h_1, h_2 were defined to be mutually exclusive, above. However, there is wiggle room here for more different kinds of games.

- The notion of Bayesian priors is optional, I think. I am defining it because I think I will need this later in this text. Other games include $\psi : H \to \mathbb{C}$ and so we call ψ a "wave function". For n qubits, one could have $\psi : H \to \mathbb{CP}^n$ where $\mathbb{CP}^n$ is complex projective space. I believe this captures the entirety of the so-called "quantum computing" game (if I'm wrong, tell me how.). The geometric computing guys write $\psi : H \to X$ where X is a homogeneous space (a topological space that is a quotient of Lie groups) and state transitions are given by the transitive actions of some finite subset of points in a (Lie) group G. As far as I can tell, "quantum computing" is a special case where $X = \mathbb{CP}^n$ (and the halting states are "measurements" i.e. projections to an orthogonal basis.) As far as I can tell, a "probabilistic state machine" is the special case where $X = \mathbb{P}(n) = \mathbf{2}^n$ the power set of n states (states of a finite machine). Likewise, a "deterministic finite automaton" has $X = \mathbf{n}$ (the set of n states). I have read texts which stated my last four sentences explicitly. I have not read texts which explicitly formulated "state transition systems" in terms of such actions of groups on homogeneous spaces, although this is implicit when one speaks of "categories of Acts", etc. I have not read about pi calculus, but I assume that these notions are implicit if not explicit. But all this is just a digression, and it might not be a useful digression. Lets continue on ...

Pause: An example

Lets pause and look at a few examples, before continuing. An olde-fashioned, 1950's thermostat consisted of a temperature sensor S generating temperature readings $i \in I$ from a set I of all possible temperature readings. (Perhaps I is the same as H above? I don't know, I am confused.) Perhaps I should use a time-label subscript i_t to denote the temperature reading at time t. However, 1950's thermostats do not include time as a part of their world-model. The progression of time is an idea in the mind of the engineer creating this particular type of thermostat. Modern thermostats do include a notion of time.

I will (provisionally?) write $S : W \to I$ so that the sensor S observes the external world W and generates a reading $i \in I$.

The reading $i \in I$ is placed in to the database B. I cannot think of any good notation for "place value into database". (Any suggestions?) There might be computation involved here, so there is some function $f : I \times B \to B$ with $f : (i,b) \mapsto b'$ that takes the pair (i,b) and performs a computation f to obtain b'. So I guess perhaps f is a state transition function? I don't yet know if this is a good way of thinking/talking about it. This is all provisional. At any rate, the idea here is that new sensory input causes the working memory B to be updated.

For the 1950's thermostat, the database B can hold only two values: "too hot" or "too cold". It is a single bit.

Next, there is an actuator $A : B \to D$ where D is a device that acts upon the world. For example, D is a heater; it is the thing that the thermostat is controlling. Here, $D = \{\text{on}, \text{off}\} = \{d_s | s = \text{device state}\} = \left\{d_{\text{on}}, d_{\text{off}}\right\}$. For each device state s, we have that $d_s : W \to W$ with $d_s : w \mapsto w'$ and this is what we mean by "the device acts upon the (external) world." Obviously, d_{on} heats the world, and d_{off} can be taken to be the identity function on the world: the no-op that does nothing.

Philosophical digression

Here, we bump heads with philosophy. From the point of view of "the good regulator", the external world W is unknown, unknowable, ineffable, except by means of the temperature sensor S. The "external world" W "exists" only in the mind of the engineer (you and I), who imagine (hallucinate) that there is such a place. The reality is that we sit behind eyeballs and ears, and have no direct access to W. We cannot "prove" it exists, which is why solipsism is an admissible philosophical position. It is also acceptable to "believe" that the "external world" exists, "in reality".

I raised this philosophical issue for a reason: We do not know, we cannot know, whether our heater is actually making an impact upon the external world. For starters, the heater might be broken: we turn it on and off (we believe it to be on and off), but nothing happens. How do we know nothing happens? Well, the temperature sensor is always giving us a temperature reading. This reading has both a random and a diurnal (24-hour) variation, in addition to whatever changes result from the heater. But what are those changes? Perhaps my heater is weak and slow. Perhaps someone opened a window. Perhaps I know the heater is broken, because the temperature no longer changes the way it used to change, last week. However, a 1950's thermostat does not

know how the temperature changed last week. It's working memory B holds only a single bit of information. It does not detect the possibility that it might be broken.

The sensory device may also be broken: it may be "hallucinating" the temperature. A time-series of temperature measurements might be generated, but these might be independent of the "actual" temperature of W (whatever the heck that is? What's the "objective reality" here?) The sensor S is taking "some reading" of W, but what that might be is only known to the engineer who designed the thermostat. It is fundamentally unknowable to the thermostat itself.

All of this whinging is to say that we don't really know what W is, we don't really know quite what it is we are perceiving, if those perceptions match "reality", or if the sensors are "hallucinating". Our actuator might be broken, and might not be having any "actual" effect upon the external universe. And we might be "crazy", in that we believe that our actions are having an effect that they do not actually have. (e.g. belief in synchronicity, magic, ESP, telekinesis, *etc.*) That is, our "world model" may be incorrect, incoherent, insane, inconsistent, in addition to merely being incomplete.

The philosophical problem here is that the external world W is, in a certain way "ineffable". I am raising this as an issue, because, mathematically, symbolically, I am using a symbol W and I am using other symbols S and D that are functions involving W. Such symbols allow me to state an algebra, and perform logical reasoning about this symbolic system. However, the foundations are shaky, because it is unclear to what extent W "actually exists".

Perhaps we can settle on a operational definition. In the mind of the engineer who is designing the thermostat, there does exist a room having a temperature, and there does exist a sensor and a heater, and these range over a set of possible values, and we can use mathematical notation to denote ("denotational semantics") these sets and functions. Insofar as all engineers are humans, there is no particular problem with partitioning W in this way, as "that thing which exists inside the engineer's head". We can make forward progress mathematically, notationally, algebraically. It is socially acceptable to pretend that W is a room in a house, and that S is a temperature sensor in that room. It is socially acceptable to solve mathematical engineering problems in this way.

Still, I am uneasy, because "denotational semantics". This W is that which only exists inside of my head (and also yours, since I believe that you exist, and I hope you read this, and I hope you understand what I meant to say.) However, I might be crazy: you might not actually exist. My hopes may be dashed: you will not heave read this far. My output device might be broken: the stuff I write might fail to have the desired effect upon the universe. My input device might be broken, and I am "actually" asleep and dreaming.

Perhaps these last seven paragraphs were silly, since obviously, every sane mathematician and engineer knows these things and take them for granted. However, in my experience of the world, I witness varying degrees of sanity, and so I felt compelled to state the obvious. I should also hand-wave "blah blah blah AGI" and "blah blah blah, Bostrom's simulation argument" just to drive the point home. Anyway ...

Recap of example

The 1950's single-bit thermostat R consists of:

- A temperature sensor $S : W \to I$ (giving a numeric temperature.)
- A working-memory update function $f : I \times B \to B$ (updating working memory to be either "too hot" or "too cold".)
- An actuator device $A : B \to D$ which selects states $d_s \in D$ which all have the form $d_s : W \to W$. Here, s is on or off, and d_s is the heater that is on or off.

That is, the regulator R is the tuple $R = (I, B, D; S, f, A)$ and for better or for worse, I left out W from this tuple, because of the philosophical observations above. Perhaps this is a mistake, and W should be a part of this tuple. Onward through the fog.

I claim that the regulator R satisfies the "good regulator theorem". From what I can tell, this theorem is not an actual mathematical theorem. But perhaps I am mistaken. Let me continue less rigorously: R satisfies the "good regulator theorem" because it contains a single-bit model of the universe: "too hot" or "too cold". But also R has an implicit meta-theory: the universe has a temperature that can be sensed, and this temp can be altered with heaters. This meta-theory exists in the engineer's head, and is used to drive engineering decisions. The meta-theory is used by the engineer to design thermostats.

Models of the World

Finally, I am able to arrive at an informal but important concept:

- T ::= a theory of the external world. This is a meta-theory, in the above sense.

I wish to be able to say that $R \in T$ or something similar. But I am confused. Is T the space of all possible thermostats? Because modern engineers can build thermostats with day/night/weekend settings. Thermostats that can record time-series of temperatures into databases B having sizes of kilobytes or megabytes. They might contain algorithms to perform diurnal analysis so that the heater comes on, on cold days, half an hour before you wake up. Some thermostats disable air-conditioning for the entire heating season of many months. A genius thermostat might attempt to predict temperatures for the next one-hundred years, although most human engineers would call such a home thermostat "insane". It is not "insane" per the earlier definitions of sanity, but because the living room that the thermostat is controlling might not exist in 100 years, and so contemplating temperatures that far in the future seems pointless and futile to mid-wit intelligences. A normie engineer will not build a consumer-grade home thermostat that remembers and ponders temperature unto eternity.

All of this verbal spillage is to drive home the idea that B alone is not the model of the world. B was working memory, a scratch pad. The database B might be organized to contain Bayesian priors, and it might be organized to perform Bayesian updates. That is, the update function $f : I \times B \to B$ might involve a Bayesian update. The update function might involve a quantum computer. The database B might be digital, storing bits, or it might store qubits. The database B may be "uncountably large", in that it

might be some distribution on a homogeneous space; it might be some Hilbert space of functions on top of a homogeneous space. The objects of B might be loops and suspensions or Leray spectral sequences or other concepts from algebraic topology. The point here is that these, umm, things, umm, objects are only "models of the world" in the small sense (of being like a stored temperature) but not in the big sense (of being like a thermostat/heater).

Some people call B the "model of the world" and some people call T the "model of the world", and the failure to clearly distinguish these two is a failure mode for professional communications.

The distinction between B and T seems to be important for defining what "learning" is.

More or less everything I have written above is conventional knowledge or standard knowledge that is well-known to practitioners versed in the state of the art. The design and analysis of regulators R is called "control theory" and there are hundreds of textbooks written about control theory. Nothing of what I've written so far should be controversial, and all of it should be find-able in some text or another.

Agents

In the above, I developed a notion of a regulator R being the tuple $R = (I, B, D; S, f, A)$. As I have not yet read or pondered the rho-calculus papers, I do not know how to convert this into rho language. I don't think it should be hard, and that exercise should reveal how to generalize my definitions.

And yet, I'm already confused. Take, for example, the sensor function $S : W \to I$. In the rho calculus, the elements of I are agents (since the rho calculus is recursive in this way) and S seems to be almost like message passing. I wrote "almost like", because it is not obvious to me that one can treat W as a collection of "external agents", from which one can be selected and then passed to the "primary agent", which receives that message. As discussed before, the "external world" W is in some sense unknowable, ineffable and unalterable except through the sensor S and the motor A, and these may be hallucinating or broken.

That is, from the "inside" of the agent, the agent "cannot know" if it is broken. The agent can have a "subjective experience" that is compatible with "functioning normally", and the brokenness is visible only to the outside observer.

I don't know how the rho calculus deals with this. Perhaps it doesn't. This is OK if you are an engineer, and it is indeed very useful to pretend that there is a "real" W out there that can be measured and acted upon. Perhaps we are even forced to be engineers, and, in the short term, can only build AGI according to the assumption that W is "real". But one can get "really out there" in physics, by hand-waving mumbo-jumbo about QM, Planck length, ER=EPR and whatnot. Assuming a Cartesian existence for W is perhaps unavoidable at this stage of the game, but I think it would be better to avoid a dependence on such an assumption.

Let me repeat that last sentence: Can we build a formal definition of a regulator that does not require specification of W? (It is OK to say "W exists", but it is not OK to say "W consists of foo, bar, baz".)

Learning

Let's now attempt to define what learning is. In what follows, I recapitulate the conventional world-view of machine-learning. I don't believe any of it is idiosyncratic or non-standard, although I do have to adapt it to the present conversation. So again, just textbook stuff.

A "learning system" is a system that creates a sequence of update functions $f_k : I \times B \to B$ with $k \in \mathbb{Z}$ and also actuators $A_k : B \to D$ such that each of these is somehow "better" than the one before. This might require a sequence of B_k to be generated. It will often be the case that $B_k \subseteq B_{k+1}$ but not always: learning might involve the discovery of a shorter "description length", which is often taken to be a "better model of the world". The words "description length" usually mean "the size of B" or perhaps "the size of $b \in B$" and so the "minimum description length" is commonly understood to be "the smallest possible size for $b \in B$ that still accurately represents the external phenomenon in W".

For example, a temporal sequence of temperature readings might be that "external phenomenon in W". The flaw is now obvious: the temperature readings are from the sensor S and we make a category error in assuming they "exist" in W. Some philosophers of AI fail to make this distinction during drunken conversations.

I am going to point out, and then ignore the possibility that there might be a sequence of I_k and of D_k. A learner might be able to build a better sensor (e.g. a human might build a microscope or telescope, to make observations the naked eye cannot make, by itself.) A learner might build a better control device (*e.g.* a human might build a motor, a steam engine, that far exceeds the strength and power of human muscles.) Talking about the learning of improved sensors and motors would seem to complicate the discussion, so I set it aside.

Utility functions

What do we mean by "better" or "improved"? Conventional machine-learning texts offer two distinctions: "supervised learning" and "unsupervised learning". In supervised learning, there is an explicit "scoring function" aka "utility function" U that tells you how well you are doing. The higher the score, the better. Learning algos can then do "hill climbing" to improve the score. For differentiable utility functions, one can take the gradient and do "gradient descent" (after a minus sign). There's an immense zoo of algos that can be applied to any given, fixed domain.

A "fixed domain" fixes the utility function U and also fixes the input alphabet I. It is often (usually) the case that learning is restricted to $D = \varnothing$ the empty set. That is, the learning system is NOT an agent, and has no effect upon the "external world". Things like LLM's and GPT fall into this class, insofar as the training of LLMs is done on some fixed texts. The external world W on which an LLM is trained is the collection of training texts. These are static, unalterable. They are held on a disk drive, they are not dynamic. Training is not interactive, and the part where humans "talk to" GPT is outside of the learning iterator.

Supervised learning

Lets try to formalize supervised learning. It requires (*ad hoc*) a utility function $U : I \times B \to \mathbb{R}$ which is able to directly access the entire "training corpus" I as well as access the entire "database of knowledge" B and assign a numeric score $u \in \mathbb{R}$. Here, we talk about I instead of the sensor $S : W \to I$ so that we can avoid talking about sensors. This is convenient and makes sense when $D = \varnothing$.

An example. For GPT, someone digitized millions of telephone conversations and printed books and created a text file of UTF-8 text that consists of words and sentences. Perhaps they've segmented into blank-delimited words and punctuation-delimited sentences. This is all I because it is post-sensory for audio and video. Here, I is called the "text corpus" because that is what linguists call it. Its non-dynamic, non-interactive, fixed.

One can also do supervised learning with agents. Best explained by example. One has a grid, 2D or 3D, say, in Minecraft. This provides the W. One has a robot agent that can perform position measurements: thus, $S : W \to I$. The agent has a motor selector $A : B \to D$ with each motor setting $d_s \in D$ giving a $d_s : W \to W$ that alters the position of the robot in "the real world" W. The update function $f : I \times B \to B$ will typically record numbers representing it's position in the real world. More accurately, it records "where it thinks it might be in W". Update algos include "dead reckoning", which is the integration of the differential equations of motion, using e.g. alpha-beta filters, or Kalman filters. Update of location might also use GPS radios to obtain position info from W (which is then merged into the dead-reckoning integral.) The motor function is obvious: say it is a quad-copter. It flies, according to $d_s : W \to W$. The reward function for an agent is then $U : B \times D \to \mathbb{R}$. (It might not be all of $\mathbb{R}$ but just a single bit of success/fail. Learning of single-bit reward functions is, of course, difficult.)

To make this less abstract, consider the case of a quadcopter carrying a grenade in Ukraine. It's task is to kill russians, and so the reward function is "dead russian is better than live russian". (One can add "surrendered russian" or "fleeing russian", but now we quibble.) The evaluation of dead vs. alive is performed via $S : W \to I$ where S is some camera spewing pixels, perhaps a radar, perhaps a temperature sensor. Of course, $S : W \to I$ must also include sensors needed for normal operation, such as airspeed and weather and ground proximity. It also includes sensors for "internal robot state" such as battery charge or rotor RPM. The contents of B includes a terrain map, a weather forecast, a collection of Bayesian priors for the current position of the quadcopter, a record of flight data. So far, I've described only what's needed for normal flying & bombing operation. It becomes a learning problem only when one has the opportunity to repeatedly kill russians. Only then can one try to find optimal flight-path and bomb-launch solutions.

The "learning" I am trying to describe here is meant to be "autonomous learning". The optimal solutions for control problems has a huge literature, dating back almost 100 years, published under the title of "operations research". However, when an engineer looks up some optimal algorithm in a book and codes this up in software, this is NOT a part of learning. Thus, we conclude that there is a factorization of $f_k : I \times B \to B$ such that $f_k = f^{(e)} \otimes f_k^{(l)}$ where $f^{(e)}$ is some optimal control algorithm provided by the engineer (and is, by definition, immutable over time), and $f_k^{(l)}$ is the sequence of update

functions that are being learned by the (supervised) learning system.

Some AI architecture texts refer to the location where the $f_k^{(l)}$ are stored as the "procedural memory" (distinguishing from "working memory"). This distinction is useful "in practice", but problematic "in theory". Arguments and objections may arise.

In case you (or any other reader of this text) think that a one-bit utility function is too difficult to learn, let me remind that DeepMind (and other systems) are able to learn how to play video games with one-bit reward functions (you won/lost the game). State-of-the-art research includes how to learn, minimizing the number of times that you loose. That is, risk-averse algorithms: one does not wish to train self-driving autonomous car systems with algorithms that frequently kill humans.

Since I have not yet read or pondered the rho calculus papers, I am now at an impasse of what to write down next. Recall we started with a regulator R being the tuple $R = (I, B, D; S, f, A)$ and we replace this with a sequence of regulators $R_k = (I, B_k, D; S, f_k, A)$. A "learning algorithm" is a function $L : R_k \mapsto R_{k+1}$ such that $U(R_{k+1}) > U(R_k)$.

Battlefield State of the Art

A propos to nothing at all. Battlefield learning: I was once hired to create a learning system that could figure out coordinated platoon movements from radar data. The radar data included moving animals (coyotes, cows), vehicles (tractors, cars, on and off the road), brightness (an ambulating farmer has less of a radar reflection than a combatant carrying a metal gun or other radar-reflecting metal items.) The goal of learning was to determine if the movements were coordinated (e.g. by radio-telephone) and what the movement was (are they gathering together, spreading apart, reconnoitering, retreating in disarray, running for the hail-mary 80 yard pass five seconds before the end of the fourth quarter.) Observe that this system can be used to analyze football games and ballet dances, and not just food-soldier tactics. So, in principle, it is data-agnostic. But it does give a flavor for the state of the art. Why am I mentioning this? I dunno. It popped into my mind. P.S. I failed, and my employment was swiftly terminated. I mention this so as not to sully your consciousness. National security trumps social justice.

Unsupervised learning

Standard machine learning texts also describe "unsupervised learning". However, the distinction between "supervised" and "unsupervised" is a bit *ad hoc*, and is more of a conventional usage, rather than a precise definition. Let me present the conventional distinction.

Older, conventional learning was posed as a task applied to a table of rows and columns, of which one column was labeled as "output" and the remaining columns labeled as "input". Each row was a "datapoint" and the learning task is to find a compact but accurate function that predicts output based on the input. The utility function is one that maximizes correctness, and usually includes a term for complexity penalty. That is, given two predictors of the output, the one with the smaller complexity is more highly valued. Some notion of Kolmogorov complexity is usually used. It's not hard to

write it down. The final accuracy or "fitness" of the learned function can be evaluated on a subset of rows that were held out during training. The word "accuracy" can refer to a minimization of false-positives, or false-negatives, or a maximization of the F1-score, or a maximal area under the ROC curve, whatever. In this context, "unsupervised training" is what you did when you did not have such a tabular form.

However, for most of these non-tabular "unsupervised" cases, there is still some sort of utility score. So, for the automated learning of video-game playing, in the end, one has a feedback signal that "you won/lost the game" that can be used during training. For LLM's, such as GPT, these are generally accepted to belong to the "unsupervised" class, and the scoring function is given by how well you predicted word w_n given the prior sequence of $(w_1, w_2, \cdots, w_{n-1})$. A different variant requires the prediction of w_n given $(\cdots, w_{n-1}, w_{n+1}, \cdots)$. One can always evaluate the final accuracy by holding out some subset of the training corpus, and evaluating your model on this holdout set.

The main point here is that, even in the conventionally "unsupervised" situation, one still has some global "utility function" that one wishes to maximize.

Utility functions redux

This is one reason why the folks at MIRI seem to be so obsessed with utility functions: it is something they can hold onto, grasp, and work with. I even recall a proposal, maybe from Nick Bostrom, which proposed that "maximizing happiness" is the appropriate utility function for AGI. Of course, the doomsters are obsessed with the idea of "maximizing paperclips" as an unwanted utility function. The e/acc people talk about maximizing the future happiness(???) of all lifeforms (???) across all spacetime continua (???) at the possible expense of humans alive today (we're going to save the lives of chickens even if that means humans will starve. Or something. It's pretty crazy out there.)

There are other possibilities. Skating on thin ice, let me try to list them.

- A utility function does not exist.
- The utility function of the universe is unknowable.
- The utility function is unknowable in the narrow sense of Turning computability.
- However, Turing machines are formulated in terms of tapes and finite sets of symbols, and the definition of geometric state machines make use of homogeneous spaces and the operators on them, which are uncountable, underneath. But I guess measure theory makes them quasi-countable. At any rate, geometric machines seem to be able to touch things that Turing machines cannot, but this is controversial. Is there a utility function that is "knowable" in the sense of a quantum finite automaton, or not?
- Ever-deeper rabbit holes.

Unknowability is a dicey thing. Personally, I am (currently) a pan-psychic, and so I like to envision the "universe" as a a collection of graphs, filigreed and complex. I'm unsure about whether I mean the physical universe, or the Platonic mathematical universe. But

these graphs seem to filigree and fill everything, in some multi-fractal way, with some uncountable number of multi-fractals. Blah blah odometers and Bretteli-Vershik diagrams blah blah (blah blah Paul Cohen forcing, blah blah sigma-pi hierarchy, ineffable cardinals, Chaitin omega, blah blah). Perhaps one can measure the complexity of such systems with Tononi Phi or something similar. Should one take the Tonini Phi as the utility function, and try to maximize that? Or will that give us paperclips? No matter how I write or talk about the preceding paragraph, Joscha Bach is guaranteed to call me crazy to my face, in public. "You are making a category error", he will say. By contrast, Ben will nod his head, stroke his chin, and say "cool cool" and look off to the upper right (my upper left) and then almost immediately launch into a long speech about something distantly related, during which I lose my train of thought, and declare surrender. I re-entrain on the current conversation and try to find something clever to say. Is this a utility function? "Try to entertain Ben, while avoiding being called crazy by Joscha?" (Is this the multi-agent version of Bostrom's "maximize happiness" utility function? Or was it "maximize entertainment"? I forget. Because it seemed silly. Still ... maximize some measure of subjective experience for some collection of agents? What?)

What am I trying to say here? Its very easy: I would prefer to have a formalization of "learning" that does not require the assumption of a "utility function".

In rho calculus, you formalized the concept of an agent without the need for a "utility function". Now, a thermostat/heater is an "agent" that is "maximizing the comfort level of the air in the room" (denotational semantics). However, the thermostat does not know that it is doing this, and the concept of "maximal comfort" is only in the mind of the engineer building the thermostat, and in the mind of the consumer buying the thermostat.

Is it possible to design without knowing what one is maximizing? Just like "philosophical zombies", can one be a "zombie engineer"? Create systems without knowing the denotational semantics of what one is creating? So, category theory is it's own meta-theory. Can I talk about a learning theory that is it's own meta-theory? That is, avoid the need for talking about utility functions?

Joscha said "everything is a function" and I said "no, not everything" and I could see in the reflections of his eyes how he revised his Bayesian priors to increase the likelihood that "Linas is crazy". Brian also blurted out "everything is a function" and I replied "triangles are not functions, even though with hard work, you can find a representation for a triangle that is a function." How can I talk about learning without talking about utility functions? How can I reach Nirvana tonight?

Coda

It is impossible to say anything further that is coherent, so I will happily quote the first verse of Frank Zappa's Cosmik Debris:

> The Mystery Man came over
>
> An' he said: "I'm outta sight!"
>
> He said, for a nominal service charge

I could reach Nirvana tonight

If I was ready, willing and able

To pay him his regular fee

He would drop all the rest of his pressing affairs

And devote his attention to me

You're quite welcome to respond with the chorus. I suspect you already know the lyrics to this song, and know how that chorus goes.