New Millennium AI and the Convergence of History: Update of 2012

Jürgen Schmidhuber

The Swiss AI Lab IDSIA
Galleria 1, 6928 Manno-Lugano, Switzerland
University of Lugano & SUPSI, Switzerland
juergen@idsia.ch - http://www.idsia.ch/~juergen

Summary. Artificial Intelligence (AI) has recently become a real formal science: the new millennium brought the first mathematically sound, asymptotically optimal, *universal* problem solvers, providing a new, rigorous foundation for the previously largely heuristic field of General AI and embedded agents. There also has been rapid progress in not quite universal but still rather general and *practical* artificial recurrent neural networks for learning sequence-processing programs, now yielding state-of-the-art results in real world applications. And the computing power per Euro is still growing by a factor of 100-1000 per decade, greatly increasing the feasibility of neural networks in general, which have started to yield human-competitive results in challenging pattern recognition competitions. Finally, a recent formal theory of fun and creativity identifies basic principles of curious and creative machines, laying foundations for artificial scientists and artists. Here I will briefly review some of the new results of my lab at IDSIA, and speculate about future developments, pointing out that the time intervals between the most notable events in over 40,000 years or 2^9 lifetimes of human history have sped up exponentially, apparently converging to zero within the next few decades. Or is this impression just a by-product of the way humans allocate memory space to past events?

Note: this is the 2012 update of a 2007 publication [73]. Compare also the 2006 celebration of 75 years of AI [71].

1 Introduction

In 2003 I observed [65, 66] that each major breakthrough in computer science tends to come roughly twice as fast as the previous one, roughly matching a century-based scale: In 1623 the computing age started with the first mechanical calculator by Wilhelm Schickard (followed by machines of Pascal, 1640, and Leibniz, 1670). Roughly two centuries later Charles Babbage came up with the concept of a program-controlled computer (1834-1840). One century later, Julius Lilienfeld invented the transistor (late 1920s), and Kurt Gödel layed the foundations of theoretical computer science with his work on universal formal languages and the limits of proof and computation (1931) [20]. His results and Church's extensions thereof were reformulated by Turing in 1936 [96], while Konrad Zuse built the first working program-controlled computers (1935-1941), using the binary system of Leibniz (1701) instead of the more cumbersome decimal system used by Babbage and many others. By 1941 all the main ingredients of ‘modern’ computer science were in place. The next 50 years saw many less radical theoretical advances as well as faster and faster switches—relays were replaced by tubes by single transistors by numerous transistors etched on chips—but arguably this was rather predictable, incremental progress without earth-shaking events. Half a century later, however, Berners-Lee triggered the most recent world-changing development by creating the World Wide Web at CERN (1990).

Extrapolating the trend, we should expect the next radical change to manifest itself one quarter of a century after the most recent one, that is, before 2020, when some computers will already match brains in terms of raw computing power, according to frequent estimates based on Moore’s law, which suggests a speed-up factor of roughly 100-1000 per decade, give or take a few years. Will the remaining series of faster and faster additional revolutions converge in an *Omega point* (term coined by Pierre Teilhard de Chardin, 1916) around 2040, when individual machines will already approach the raw computing power of all human brains combined, provided Moore’s law does not break down? Many of the present readers of this article should still be alive then. Compare Stanislaw Ulam’s concept of an approaching *historic singularity* (quote: [100]), popularized by Vernor Vinge as *technological singularity* [99, 100], as well as subsequent speculations [39, 47].

Will the software and the theoretical advances keep up with the hardware development? I am convinced they will. In fact, the new millennium has brought not only human-competitive performance of artificial neural networks (NN) in pattern recognition contests (more on this later), but also fundamental new insights into the problem of constructing *theoretically optimal* rational agents or universal Artificial Intelligences (AIs), as well as curious & creative machines (more on this below). There also has been rapid progress in *practical* learning algorithms for agents interacting with a dynamic environment, autonomously discovering true sequence-processing, problem-solving programs, as opposed to the reactive mappings from stationary inputs to outputs studied in most traditional machine learning (ML) research. In what follows, I will briefly review some of the new results, then come back to the issue of whether or not history is about to “converge.”

2 Notation

Consider a learning robotic agent with a single life which consists of discrete cycles or time steps $t = 1, 2, \ldots, T$. Its total lifetime T may or may not be known in advance. In what follows,the value of any time-varying variable Q at time t ($1 \leq t \leq T$) will be denoted by $Q(t)$, the ordered sequence of values $Q(1), \ldots, Q(t)$ by $Q(\leq t)$, and the (possibly empty) sequence $Q(1), \ldots, Q(t-1)$ by $Q(< t)$.

At any given t the robot receives a real-valued input vector $x(t)$ from the environment and executes a real-valued action $y(t)$ which may affect future inputs; at times $t < T$ its goal is to maximize future success or *utility*

$$u(t) = E_\mu \left[\sum_{\tau=t+1}^{T} r(\tau) \;\middle|\; h(\leq t) \right], \tag{1}$$

where $r(t)$ is an additional real-valued reward input at time t, $h(t)$ the ordered triple $[x(t), y(t), r(t)]$ (hence $h(\leq t)$ is the known history up to t), and $E_\mu(\cdot \mid \cdot)$ denotes the conditional expectation operator with respect to some possibly unknown distribution μ from a set M of possible distributions. Here M reflects whatever is known about the possibly probabilistic reactions of the environment. For example, M may contain all computable distributions [32, 44, 88]. Note that unlike in most previous work by others [34, 95], but like in much of the author's own previous work [72, 81], there is just one life, no need for predefined repeatable trials, no restriction to Markovian interfaces between sensors and environment [58], and the utility function implicitly takes into account the expected remaining lifespan $E_\mu(T \mid h(\leq t))$ and thus the possibility to extend it through appropriate actions [68, 72].

3 Universal But Incomputable AI

Solomonoff's theoretically optimal universal predictors and their Bayesian learning algorithms [32, 44, 88] only assume that the reactions of the environment are sampled from an unknown probability distribution μ contained in a set M of all enumerable distributions—compare text after equation (1). That is, given an observation sequence $q(\leq t)$, we only assume there exists a computer program that can compute the probability of the next possible $q(t+1)$, given $q(\leq t)$. Since we typically do not know the program computing μ, we predict the future in a Bayesian framework by using a mixture distribution $\xi = \sum_i w_i \mu_i$, a weighted sum of *all* distributions $\mu_i \in \mathcal{M}$, $i = 1, 2, \ldots$, where $\sum_i w_i \leq 1$. It turns out that this is indeed the best one can possibly do, in a very general sense [32]. The drawback is that the scheme is incomputable, since M contains infinitely many distributions.

One can increase the theoretical power of the scheme by augmenting M by certain non-enumerable but limit-computable distributions [63], or restrict it such that it becomes computable, e.g., by assuming the world is computed by some unknown but deterministic computer program sampled from the Speed Prior [64] which assigns low probability to environments that are hard to compute by any method. Under the Speed Prior the cumulative a priori probability of all data whose computation through an optimal algorithm requires more than $O(n)$ resources is $1/n$.

Can we use the optimal predictors to build an optimal AI? Indeed, in the new millennium it was shown we can. At any time t, the recent theoretically optimal yet uncomputable RL algorithm AIXI [32] uses Solomonoff's universal prediction scheme to select those action sequences that promise maximal future reward up to some horizon, typically $2t$, given the current data $h(\leq t)$. One may adapt this to the case of any finite horizon T. That is, in cycle $t+1$, AIXI selects as its next action the first action of an action sequence maximizing ξ-predicted reward up to the horizon, appropriately generalizing Solomonoff's universal prior. Recent work [32] demonstrated AIXI's optimal use of observations as follows. The Bayes-optimal policy p^ξ based on the mixture ξ is self-optimizing in the sense that its average utility value converges asymptotically for all $\mu \in \mathcal{M}$ to the optimal value achieved by the (infeasible) Bayes-optimal policy p^μ which knows μ in advance. The necessary condition that $\mathcal{M}$ admits self-optimizing policies is also sufficient. Furthermore, p^ξ is Pareto-optimal in the sense that there is no other policy yielding higher or equal value in *all* environments $\nu \in \mathcal{M}$ and a strictly higher value in at least one [32].

What are the implications? The first decades of attempts at *Artificial General Intelligence (AGI)* have been dominated by heuristic approaches [46, 48, 52, 97]. Traditionally many theoretical computer scientists have regarded the field with contempt for its lack of hard theoretical results. Things have changed, however. Although the universal approach above is practically infeasible due to the incomputability of Solomonoff's prior, it does provide, for the first time, a mathematically sound theory of AGI and optimal decision making based on experience, identifying the limits of both human and artificial intelligence, and providing a yardstick for any future approach to AGI.

Using the Speed Prior mentioned above, one can scale the universal approach down such that it becomes at least *computable* [64]. In what follows I will mention ways of introducing additional optimality criteria that take into account the computational costs of prediction and decision making.

4 Asymptotically Optimal General Problem Solver

To take computation time into account in a general, theoretically optimal way [43] [44, p. 502-505], the recent asymptotically optimal search algorithm for *all* well-defined problems HSEARCH [31] uses a hardwired brute force proof searcher which (justifiably) ignores the costs of proof search. Assuming discrete input/output domains $X/Y \subset B^*$, a formal problem specification $f : X \rightarrow Y$ (say, a functional description of how integers are decomposed into their prime factors), and a particular $x \in X$ (say, an integer to be factorized), HSEARCH orders all proofs of an appropriate axiomatic system by size to find programs q that for all $z \in X$ provably compute $f(z)$ within time bound $t_q(z)$. Simultaneously it spends most of its time on executing the q with the best currently proven time bound $t_q(x)$. Remarkably, HSEARCH is as fast as the *fastest* algorithm that provably computes $f(z)$ for all $z \in X$, save for a constant factor smaller than $1 + \epsilon$ (arbitrary real-valued $\epsilon > 0$) and an f-specific but x-independent additive constant [31].

Practical applications, however, should not ignore potentially huge constants. This motivates the next section which addresses all kinds of optimality (not just asymptotic optimality).

5 Optimal Self-Referential General Problem Solver

The recent Gödel machines [68, 72, 74] represent the first class of mathematically rigorous, general, fully self-referential, self-improving, optimally efficient problem solvers. In particular, they are applicable to the problem embodied by objective (1), which obviously is not limited to *asymptotic* optimality. Gödel machines formalize I. J. Good's informal remarks (1965) on an *"intelligence explosion through self-improving super-intelligences."*

The initial software $\mathcal{S}$ of such a Gödel machine contains an initial problem solver, e.g., one of the approaches above [32] or some less general, typical sub-optimal method [34, 95]. Simultaneously, it contains an initial proof searcher (possibly based on an online variant of *Universal Search* [43] or the *Optimal Ordered Problem Solver* [67]) which is used to run and test *proof techniques*. The latter are programs written in a universal programming language implemented on the Gödel machine within $\mathcal{S}$, able to compute proofs concerning the system's own future performance, based on an axiomatic system $\mathcal{A}$ encoded in $\mathcal{S}$. $\mathcal{A}$ describes the formal *utility* function, in our case eq. (1), the hardware properties, axioms of arithmetics and probability theory and string manipulation etc, and $\mathcal{S}$ itself, which is possible without introducing circularity [68, 72, 74].

Inspired by Kurt Gödel's celebrated self-referential formulas (1931) [20], the Gödel machine rewrites any part of its own code in a computable way through a self-generated executable program as soon as its *Universal Search* variant has found a proof that the rewrite is *useful* according to objective (1). According to the Global Optimality Theorem [68, 72, 74], such a self-rewrite is globally optimal—no local maxima!—since the self-referential code first had to prove that it is not useful to continue the proof search for alternative self-rewrites.

If there is no provably useful, globally optimal way of rewriting $\mathcal{S}$ at all, then humans will not find one either. But if there is one, then $\mathcal{S}$ itself can find and exploit it. Unlike *non*-self-referential methods based on hardwired proof searchers [32] (Section 4), Gödel machines not only boast an optimal *order* of complexity but can optimally reduce (through self-changes) any slowdowns hidden by the asymptotic $O()$-notation, provided the utility of such speed-ups is provable at all.

To make sure the Gödel machine is at least *asymptotically* optimal even before the first self-rewrite, we may initialize it by the non-self-referential but *asymptotically fastest algorithm for all well-defined problems* HSEARCH [31] of Section 4. Given some problem, the Gödel machine may decide to replace its HSEARCH initialization by a faster method suffering less from large constant overhead, but even if it doesn't, its performance won't be less than asymptotically optimal.

Implications

The above implies that there already exists the blueprint of a Universal AI which will solve almost all problems almost as quickly as if it already knew the best (unknown) algorithm for solving them, because almost all imaginable problems are big enough to make additive constants negligible. The only motivation for *not* quitting computer science research right now is that many real-world problems are so small and simple that the ominous constant slowdown (potentially relevant at least before the first Gödel machine self-rewrite) is *not* negligible.

6 Recurrent / Deep Neural Networks

Practical implementations of the Gödel machine above do not yet exist, and probably will require a thoughtful choice of the initial axioms and the initial proof searcher. In what follows, however, I will focus on already quite practical, non-optimal and non-universal, but still rather general searchers in program space, as opposed to the space of reactive, feedforward input / output mappings, which still attracts the bulk of current ML research.

Recurrent NN (RNN) are NN [4] with feedback connections that are, in principle, as powerful as any traditional computer. There is a very simple way to see this [55]: a traditional microprocessor may be viewed as a very sparsely connected RNN consisting of very simple neurons implementing nonlinear AND and NAND gates, etc. Compare [84] for a more complex argument. Hence RNN can solve tasks involving sequences of continually varying inputs. Examples include robot control, speech recognition, music composition, attentive vision, and numerous others.

Supervised RNN can be trained by gradient descent and other methods [33, 45, 51, 59, 101, 104]. Recent work has successfully applied Hessian-free optimization to RNN [94], using tricks such as special damping functions and stopping criteria, mini-batches for curvature calculation, and others. Our own RNN overcome fundamental problems of previous RNN [28], outperforming them in many applications [16, 17, 22–25, 29, 80]. While RNN used to be **toy problem methods** in the 1990s, ours have recently started to outperform all other methods in **challenging real world applications** [14, 23–25, 78, 80]. Recently, our CTC-trained [22] mulitdimensional [25] RNN won three Connected Handwriting Recognition Competitions at ICDAR 2009 (see list of won competitions below).

Training an RNN by standard methods is similar to training a feedforward NN (FNN) with many layers, which runs into similar problems [28]. However, our recent deep FNN with special internal architecture overcome these problems to the extent that they are currently winning many international pattern recognition contests [6, 8–11, 78] (see list of won competitions below). None of this requires the traditional sophisticated computer vision techniques developed over the past six decades or so. Instead, our biologically rather plausible NN architectures learn from experience with millions of training examples. Typically they have many non-linear processing stages like Fukushima's Neocognitron [15]; we sometimes (but not always) profit from sparse network connectivity and techniques such as weight sharing & convolution [3, 40], max-pooling [54], and contrast enhancement like the one automatically generated by unsupervised *Predictability Minimization* [60, 79, 82]. Our NN are now often outperforming all other methods including the theoretically less general and less powerful support vector machines (SVM) based on statistical learning theory [98] (which for a long time had the upper hand, at least in practice). These results are currently contributing to a second **Neural Network ReNNaissance** (the first one happened in the 1980s and early 90s) which might not be possible without dramatic advances in computational power per Swiss Franc, obtained in the new millennium. In particular, to implement and train our NN, we exploit graphics processing units (GPUs, mini-supercomputers normally used for video games) which are 100 times faster than traditional CPUs, and a million times faster than PCs of two decades ago when we started this type of research.

1st Ranks of my Lab's Methods in International Competitions since 2009:
7. ISBI 2012 Segmentation Challenge (with superhuman pixel error rate) [5].
6. IJCNN 2011 on-site Traffic Sign Recognition Competition (0.56% error rate, the only method better than humans, who achieved 1.16% on average; 3rd place for 1.69%) [10]
5. ICDAR 2011 offline Chinese handwritten character recognition competition [11].
4. Online German Traffic Sign Recognition Contest (1st & 2nd rank; 1.02% error rate) [9].
3. ICDAR 2009 Arabic Connected Handwriting Competition (won by our LSTM RNN [24, 25], same below).
2. ICDAR 2009 Handwritten Farsi/Arabic Character Recognition Competition.
1. ICDAR 2009 French Connected Handwriting Competition.

1st Ranks in Important Machine Learning (ML) Benchmarks since 2010:
3. MNIST handwritten digits data set [40] (perhaps the most famous ML benchmark). New records: 0.35% error in 2010 [6], 0.27% in 2011 [7], first human-competitive performance (0.23%) in 2012 [11].
2. NORB stereo image data set [41]. New records in 2011, 2012, e.g., [11].
1. CIFAR-10 image data set [37]. New records (eventually 11.2% error rate) in 2011, 2012, e.g., [11].

In a certain sense, **Reinforcement Learning** (RL) [34, 95] is more challenging than supervised learning as above, since there is no teacher providing desired outputs at appropriate time steps. To solve a given problem, the learning agent itself must discover useful output sequences in response to the observations. The traditional approach to RL is best embodied by Sutton and Barto's book [95]. It makes strong assumptions about the environment, such as the Markov assumption: the current input of the agent tells it all it needs to know about the environment. Then all we need to learn is some sort of reactive mapping from stationary inputs to outputs. This is often unrealistic. A more general approach for partially observable environments directly evolves programs for RNN with internal states (no need for the Markovian assumption), by applying evolutionary algorithms [30, 50, 83] to RNN weight matrices [26, 85, 89, 105]. Recent work brought progress through a focus on reducing search spaces by co-evolving the comparatively small weight vectors of individual neurons and synapses [21], by Natural Gradient-based Stochastic Search Strategies [19, 53, 92, 93, 102, 103], and by reducing search spaces through weight matrix compression [36, 61]. Our RL RNN now outperform many previous methods on benchmarks [21], creating memories of important events and solving numerous tasks unsolvable by classical RL methods. Several *best paper awards* resulted from this research, e.g., [18, 92].

7 Curious & Creative Machines Maximizing Wow-Effects

The main problem in many RL tasks, however, remains the very rare external reward. How to learn anything from such limited feedback in reasonable time? Over the past two decades I have pioneered a **Formal Theory of Curiosity** (FTC) and creativity and exploration, which describes how to provide frequent additional *intrinsic* rewards for active data-creating explorers [56, 57, 62, 69, 75, 76, 90]. FTC has recently gained a lot of traction (most citations, even of the old papers, stem from the past five years, especially the last two years).

One inspiration of FTC is biological. To solve existential problems such as avoiding hunger or thirst, a baby has to learn how the environment responds to its actions. Even when there is no immediate need to satisfy thirst or other built-in primitive drives, the baby does not run idle. Instead it actively conducts non-random experiments: what sensory feedback do I get if I move my eyes or my fingers or my tongue in particular ways? Being able to predict the effects of its actions enable it to plan control sequences leading to desirable states, such as those where its thirst and hunger sensors are switched off. But the growing infant quickly gets bored by things it already understands well as well as those it does not understand at all. It searches for new effects exhibiting some yet unexplained but *easily learnable* regularities. It continually acquires more and more complex skills building on previously acquired, simpler skills. Eventually the baby may become a physicist, creating experiments to discover previously unknown physical laws, or an artist creating new eye-opening artworks, or a comedian delighting audiences with novel jokes.

According to FTC [56, 57, 62, 69, 75, 76, 90], the baby's exploratory behavior is driven by a very simple algorithmic mechanism that uses RL to maximize internal *wow effects*. Wow effects are sudden reductions in an agent's estimate of the complexity of its history of observations and actions. These occur due to the agent's own *learning progress*. To clarify, consider an explorer with two modules: a world model and an actor. The former encodes the agent's growing history of sensory data (tactile, auditory, visual, etc), while the latter executes actions that influence and shape that history. The world model (e.g., an NN or RNN) uses a learning algorithm to encode the data more efficiently, trying to discover *new regularities* that allow for saving storage space (e.g., synapses) or computation time. When successful, the RL actor receives a reward (the *wow effect*). Maximizing future expected reward, the actor is motivated to invent behaviors leading to more such rewards; i.e., to data that the encoder does not yet know but can easily learn. Wow effects can also result from simplifying or speeding up the actor itself [76]. Unlike the pre-programmed interestingness measure of EURISKO [42], FTC's continually redefines what's interesting based on what's currently easy to learn, in addition to what's already known.

Since 1990 we have been building explorers based on FTC. These agents may be viewed as simple artificial scientists or artists with an intrinsic desire to create experiments for building better models of the world [56, 57, 62, 69, 75, 90], in the process developing more and more efficient procedures or skills [62, 76]. This work has inspired much recent research; the last few years brought lots of related work by others [2, 12, 49, 86], also in the nascent field of developmental robotics [27, 38, 49]. FTC generalizes *active learning* [1, 13, 91], taking into account: (1) highly environment-dependent costs of obtaining or creating not just individual data points but data *sequences* of unknown size; (2) arbitrary algorithmic [35, 44, 63, 64, 88] or statistical dependencies in sequences of actions and sensory inputs [62, 69]; and (3) the computational cost of learning new skills [62, 76].

The first curious explorers from the 1990s [56, 62, 90] used RL methods that were sub-optimal for online learning and for *wow effect* rewards that vanish as soon as learning progress stops. More recent, mathematically optimal, creative explorers [69, 75] are based on universal RL methods [32, 64, 70] that are not yet computationally tractable (Section 3). Recent work has demonstrated exploration that is both optimal *and* feasible [106] for limited scenarios, but much remains to be done for challenging, high-dimensional, partially observable worlds. This is driving ongoing work.

8 Is History Converging? Again?

Many predict that within a few decades there will be computers whose raw computing power will surpass the one of a human brain by far (e.g., [39, 47]). In the 1980s, an educated guess of this type motivated me to study computer science and AI. I have argued above that algorithmic advances are keeping up with the hardware development, pointing to new-millennium theoretical insights on universal problem solvers and creative machines that are optimal in various mathematical senses (thus making *General AI* a real formal science), as well as to practical progress in program learning through brain-inspired neural nets.

A single human predicting the future of humankind is like a single neuron predicting what its brain will do. Nevertheless, a few things can be predicted confidently, such as: tomorrow the sun will shine in the Sahara desert. So let us put the AI-oriented developments discussed above in a broader context, and try to extend the naive analysis of past computer science breakthroughs in the introduction, which predicts that computer history will converge in an *Omega point* or historic singularity Ω around 2040 [71, 73].

Surprisingly, even if we go back all the way to the beginnings of modern man over 40,000 years ago, essential historic developments (that is, the subjects of the major chapters in history books) match a a binary scale marking exponentially declining temporal intervals, each half the size of the previous one, and even measurable in terms of powers of 2 multiplied by a human lifetime (roughly 80 years—throughout recorded history many individuals have reached this age, although the average lifetime often was shorter, mostly due to high children mortality). Using the value $\Omega = 2040$, associate an error bar of not much more than 10 percent with each date below:

1. $\Omega - 2^9$ lifetimes: modern humans start colonizing the world from Africa
2. $\Omega - 2^8$ lifetimes: bow and arrow invented; hunting revolution
3. $\Omega - 2^7$ lifetimes: invention of agriculture; first permanent settlements; beginnings of civilization

4. $\Omega - 2^6$ lifetimes: first high civilizations (Sumeria, Egypt), and the most important invention of recorded history, namely, the one that made recorded history possible: writing
5. $\Omega - 2^5$ lifetimes: The *Axial Age* (the axis around which history turned, according to Karl Jaspers), the age of the first large empire (the Persian one), the only empire ever to contain almost half humankind. At its fringes, the ancient Greeks invent democracy and lay the foundations of Western science and art and philosophy, from algorithmic procedures and formal proofs to anatomically perfect sculptures, harmonic music, sophisticated machines including steam engines, and organized sports. Major Asian religions founded, Old Testament written (basis of Judaism, Christianity, Islam). High civilizations in China, origin of the first calculation tools, and India, origin of alphabets and the zero.
6. $\Omega - 2^4$ lifetimes: bookprint (often called the most important invention of the past 2000 years) invented in China. Islamic science and culture start spreading across large parts of the known world (this has sometimes been called the most important development between Antiquity and the age of discoveries)
7. $\Omega - 2^3$ lifetimes: the most dominant empire of the past 2,500 years (the Mongolian empire) includes most of the civilized world. Soon afterwards, Chinese fleets and later also European vessels start exploring the world. Gun powder and guns invented in China. Rennaissance and printing press (often called the most influential invention of the past 1000 years) and subsequent Reformation in Europe. Begin of the Scientific Revolution
8. $\Omega - 2^2$ lifetimes: Age of enlightenment and rational thought in Europe. Massive progress in the sciences; first flying machines; start of the industrial revolution based on improved steam engines
9. $\Omega - 2$ lifetimes: Birth of the modern world in the second industrial revolution based on combustion engines, cheap electricity, and modern chemistry. Genetic and evolution theory. Revolutionary modern medicine through the germ theory of disease. Onset of the unprecedented population explosion driving many other developments. European colonialism at its short-lived peak
10. $\Omega - 1$ lifetime: Post-World War II society and pop culture emerges. The worldwide super-exponential population explosion (mainly due to the Haber-Bosch process [87]) is at its peak. First commercial computers and first spacecraft; DNA structure unveiled
11. $\Omega - 1/2$ lifetime: 3rd industrial revolution (?) through an emerging worldwide digital nervous system based on personal computers, cell phones, and the World Wide Web. A mathematical theory of universal AI emerges (see sections above)—will this be considered a milestone in the future?
12. $\Omega - 1/4$ lifetime: This point will be reached in a few years. See introduction
13. $\Omega - 1/8$ lifetime: The number of humans will roughly match the number of grey matter neurons in a human brain. Will they be digitally connected in a roughly brain-like way (on average 10,000 connections per unit, mostly between neighbors arranged in a two-dimensional sheet), like a super-brain whose super-neurons are standard human brains? 100 years after Gödel's paper on the limits of proof & computation & AI [20]: will practical variants of Gödel ma-

chines start a runaway evolution of continually self-improving superminds way beyond human imagination, causing far more unpredictable revolutions in the final decade before Ω than during all the millennia before?
14. ...

I feel there is no need to justify a much more cautious outlook by pessimistically referring to comparatively recent over-optimistic and self-serving predictions (1960s: "only 10 instead of 100 years needed to build AIs") by a few early AI enthusiasts in search of funding [77]. Nevertheless, after 10,000 years of civilization it would not matter much if the Ω estimate above were off by a few decades. Note that by cosmic standards the invention of writing over 5000 years ago almost coincided with the emergence of the WWW, and all of civilization history seems like a sudden flash—one needs to zoom in very closely to resolve the minute details of this ongoing turbulent intelligence explosion spanning just a few millennia.

The following disclosure should help the reader to take the list above with a grain of salt though. I admit being very interested in witnessing the Omega point. I was born in 1963, and therefore perhaps should not expect to live long past 2040. This may motivate me to uncover certain historic patterns that fit my desires, while ignoring other patterns that do not.

Others may feel attracted by the same trap, identifying exponential speedups in sequences of historic paradigm shifts identified by various historians, to back up the hypothesis that *Omega is near*, e.g., [39]. The cited historians are all contemporary, presumably being subject to a similar bias. People of past ages might have held quite different views. For example, possibly some historians of the year 1525 felt inclined to predict a convergence of history around 1540, deriving this date from an exponential speedup of recent breakthroughs such as the printing press (around 1444), the re-discovery of America (48 years later), the Reformation (again 24 years later—see the pattern of exponential acceleration?), and other events they deemed important although today they are mostly forgotten. (According to TIME LIFE magazine's millennium issue, the three events above were the previous millennium's most influential ones.)

Could it be that such lists just reflect the human way of allocating memory space to past events [71, 73]? Maybe there is a general rule for both the individual memory of single humans and the collective memory of entire societies and their history books: constant amounts of memory space get allocated to exponentially larger, adjacent time intervals further and further into the past. For example, events that happened between 2 and 4 lifetimes ago get roughly as much memory space as events in the previous interval of twice the size. Presumably only a few "important" memories will survive the necessary compression. Maybe that's why there has never been a shortage of prophets predicting that the end is near—the important events according to one's own view of the past always seem to accelerate exponentially. A similar plausible type of memory decay allocates $O(1/n)$ memory units to all events older than $O(n)$ unit time intervals. This is reminiscent of a bias governed by a time-reversed Speed Prior [64] (Section 3).

References

[1] M. F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *J. Comput. Syst. Sci.*, 75(1):78–89, 2009.

[2] A. Barto. Intrinsic motivation and reinforcement learning. In G. Baldassarre and M. Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2012. In press.

[3] S. Behnke. *Hierarchical Neural Networks for Image Interpretation*, volume 2766 of *Lecture Notes in Computer Science*. Springer, 2003.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] D. C. Ciresan, A. Giusti, and J. Schmidhuber. Neural networks for segmenting neuronal structures in EM stacks. In *ISBI Segmentation Challenge Competition: Abstracts*, 2012, in press.

[6] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep big simple neural nets for handwritten digit recogntion. *Neural Computation*, 22(12):3207–3220, 2010.

[7] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In *11th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1250–1254, 2011.

[8] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Intl. Joint Conference on Artificial Intelligence IJCAI*, pages 1237–1242, 2011.

[9] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. In *International Joint Conference on Neural Networks*, pages 1918–1921, 2011.

[10] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, in press, 2012.

[11] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2012*, 2012, in press. Long preprint arXiv:1202.2745v1 [cs.CV].

[12] P. Dayan. Exploration from generalization mediated by multiple controllers. In G. Baldassarre and M. Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2012. In press.

[13] V. V. Fedorov. *Theory of optimal experiments*. Academic Press, 1972.

[14] S. Fernandez, Alex Graves, and J. Schmidhuber. Sequence labelling in structured domains with hierarchical recurrent neural networks. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[15] K. Fukushima. Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[16] F. A. Gers and J. Schmidhuber. LSTM recurrent networks learn simple context free and context sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001.

[17] F. A. Gers, N. Schraudolph, and J. Schmidhuber. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 3:115–143, 2002.

[18] L. Gisslen, M. Luciw, V. Graziano, and J. Schmidhuber. Sequential constant size compressor for reinforcement learning. In *Proc. Fourth Conference on Artificial General Intelligence (AGI), Google, Mountain View, CA*, 2011.

[19] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2010.

[20] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[21] F. J. Gomez, J. Schmidhuber, and R. Miikkulainen. Efficient non-linear control through neuroevolution. *Journal of Machine Learning Research JMLR*, 9:937–965, 2008.

[22] A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural nets. In *ICML '06: Proceedings of the International Conference on Machine Learning*, 2006.

[23] A. Graves, S. Fernandez, M. Liwicki, H. Bunke, and J. Schmidhuber. Unconstrained on-line handwriting recognition with recurrent neural networks. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 577–584. MIT Press, Cambridge, MA, 2008.

[24] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 2009.

[25] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems 21*. MIT Press, Cambridge, MA, 2009.

[26] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[27] S. Hart, S. Sen, and R. Grupen. Intrinsically motivated hierarchical manipulation. In *Proceedings of the 2008 IEEE Conference on Robots and Automation (ICRA)*, Pasadena, California, 2008.

[28] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.

[29] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[30] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[31] M. Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(3):431–443, 2002. (On J. Schmidhuber's SNF grant 20-61847).

[32] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005. (On J. Schmidhuber's SNF grant 20-61847).

[33] H. Jaeger. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304:78–80, 2004.

[34] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of AI research*, 4:237–285, 1996.

[35] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–11, 1965.

[36] J. Koutnik, F. Gomez, and J. Schmidhuber. Evolving neural networks in compressed weight space. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO-10)*, 2010.

[37] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Computer Science Department, University of Toronto, 2009.

[38] B. Kuipers, P. Beeson, J. Modayil, and J. Provost. Bootstrap learning of foundational representations. *Connection Science*, 18(2), 2006.

[39] R. Kurzweil. *The Singularity is near*. Wiley Interscience, 2005.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[41] Yann LeCun, Fu-Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. of Computer Vision and Pattern Recognition Conference*, 2004.

[42] D. B. Lenat. Theory formation by heuristic search. *Machine Learning*, 21, 1983.

[43] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[44] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (2nd edition)*. Springer, 1997.

[45] W. Maass, T. Natschläger, and H. Markram. A fresh look at real-time computation in generic recurrent neural circuits. Technical report, Institute for Theoretical Computer Science, TU Graz, 2002.

[46] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[47] H. Moravec. *Robot*. Wiley Interscience, 1999.

[48] A. Newell and H. Simon. GPS, a program that simulates human thought. In E. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 279–293. McGraw-Hill, New York, 1963.

[49] P.-Y. Oudeyer, A. Baranes, and F. Kaplan. Intrinsically motivated learning of real world sensorimotor skills with developmental constraints. In G. Baldassarre and M. Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2012. In press.

[50] I. Rechenberg. Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dissertation, 1971. Published 1973 by Fromman-Holzboog.

[51] A. J. Robinson and F. Fallside. The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.

[52] P. S. Rosenbloom, J. E. Laird, and A. Newell. *The SOAR Papers*. MIT Press, 1993.

[53] T. Schaul, J. Bayer, D. Wierstra, Yi Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

[54] D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, 2010.

[55] J. Schmidhuber. Dynamische neuronale Netze und das fundamentale raumzeitliche Lernproblem. Dissertation, Institut für Informatik, Technische Universität München, 1990.

[56] J. Schmidhuber. Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press, 1991.

[57] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson, editors, *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227. MIT Press/Bradford Books, 1991.

[58] J. Schmidhuber. Reinforcement learning in Markovian and non-Markovian environments. In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3 (NIPS 3)*, pages 500–506. Morgan Kaufmann, 1991.

[59] J. Schmidhuber. A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks. *Neural Computation*, 4(2):243–248, 1992.

[60] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[61] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

[62] J. Schmidhuber. Artificial curiosity based on discovering novel algorithmic predictability through coevolution. In P. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and Z. Zalzala, editors, *Congress on Evolutionary Computation*, pages 1612–1618. IEEE Press, 1999.

[63] J. Schmidhuber. Hierarchies of generalized Kolmogorov complexities and nonenumerable universal measures computable in the limit. *International Journal of Foundations of Computer Science*, 13(4):587–612, 2002.

[64] J. Schmidhuber. The Speed Prior: a new simplicity measure yielding near-optimal computable predictions. In J. Kivinen and R. H. Sloan, editors, *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 2002)*, Lecture Notes in Artificial Intelligence, pages 216–228. Springer, Sydney, Australia, 2002.

[65] J. Schmidhuber. Exponential speed-up of computer history's defining moments, 2003. http://www.idsia.ch/˜juergen/computerhistory.html.

[66] J. Schmidhuber. The new AI: General & sound & relevant for physics. Technical Report TR IDSIA-04-03, Version 1.0, arXiv:cs.AI/0302012 v1, February 2003.

[67] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.

[68] J. Schmidhuber. Completely self-referential optimal reinforcement learners. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, editors, *Artificial Neural Networks: Biological Inspirations - ICANN 2005, LNCS 3697*, pages 223–233. Springer-Verlag Berlin Heidelberg, 2005. Plenary talk.

[69] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[70] J. Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, pages 199–226. Springer Verlag, 2006. Variant available as arXiv:cs.LO/0309048.

[71] J. Schmidhuber. 2006: Celebrating 75 years of AI - history and outlook: the next 25 years. In M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, editors, *50 Years of Artificial Intelligence*, volume LNAI 4850, pages 29–41. Springer Berlin / Heidelberg, 2007. Preprint available as arXiv:0708.4311.

[72] J. Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, pages 199–226. Springer Verlag, 2007. Variant available as arXiv:cs.LO/0309048.

[73] J. Schmidhuber. New millennium AI and the convergence of history. In W. Duch and J. Mandziuk, editors, *Challenges to Computational Intelligence*, volume 63, pages 15–36. Studies in Computational Intelligence, Springer, 2007. Also available as arXiv:cs.AI/0606081.

[74] J. Schmidhuber. Ultimate cognition *à la* Gödel. *Cognitive Computation*, 1(2):177–193, 2009.

[75] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[76] J. Schmidhuber. POWERPLAY: Training an Increasingly General Problem Solver by Continually Searching for the Simplest Still Unsolvable Problem. Technical Report arXiv:1112.5309v1 [cs.AI], 2011.

[77] J. Schmidhuber. Philosophers & futurists, catch up! Response to the singularity. *Journal of Consciousness Studies*, 19(1-2):173–182, 2012.

[78] J. Schmidhuber, D. Ciresan, U. Meier, J. Masci, and A. Graves. On fast deep nets for AGI vision. In *Proc. Fourth Conference on Artificial General Intelligence (AGI), Google, Mountain View, CA*, 2011.

[79] J. Schmidhuber, M. Eldracher, and B. Foltin. Semilinear predictability minimization produces well-known feature detectors. *Neural Computation*, 8(4):773–786, 1996.

[80] J. Schmidhuber, D. Wierstra, M. Gagliolo, and F. J. Gomez. Training recurrent networks by EVOLINO. *Neural Computation*, 19(3):757–779, 2007.

[81] J. Schmidhuber, J. Zhao, and N. Schraudolph. Reinforcement learning with self-modifying policies. In S. Thrun and L. Pratt, editors, *Learning to learn*, pages 293–309. Kluwer, 1997.

[82] N. N. Schraudolph, M. Eldracher, and J. Schmidhuber. Processing images by semi-linear predictability minimization. *Network: Computation in Neural Systems*, 10(2):133–169, 1999.

[83] H. P. Schwefel. Numerische Optimierung von Computer-Modellen. Dissertation, 1974. Published 1977 by Birkhäuser, Basel.

[84] H. T. Siegelmann and E. D. Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80, 1991.

[85] K. Sims. Evolving virtual creatures. In A. Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 1994)*, Computer Graphics Proceedings, Annual Conference, pages 15–22. ACM SIGGRAPH, ACM Press, jul 1994. ISBN 0-89791-667-0.

[86] S. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 (NIPS)*. MIT Press, Cambridge, MA, 2005.

[87] V. Smil. Detonator of the population explosion. *Nature*, 400:415, 1999.

[88] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7:1–22, 1964.

[89] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

[90] J. Storck, S. Hochreiter, and J. Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks, Paris*, volume 2, pages 159–164. EC2 & Cie, 1995.

[91] A. Strehl, J. Langford, and S. Kakade. Learning from logged implicit exploration data. Technical Report arXiv:1003.0120, Mar 2010.

[92] Yi Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient natural evolution strategies. In *Genetic and Evolutionary Computation Conference*, 2009.

[93] Yi Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic search using the natural gradient. In *International Conference on Machine Learning (ICML)*, 2009.

[94] I. Sutskever, J. Martens, and G. Hinton. Generating text with recurrent neural networks. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1017–1024, New York, NY, USA, June 2011. ACM.

[95] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press, 1998.

[96] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2*, 41:230–267, 1936.

[97] P. Utgoff. Shift of bias for inductive concept learning. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning*, volume 2, pages 163–190. Morgan Kaufmann, Los Altos, CA, 1986.

[98] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[99] V. Vinge. *The Peace War*. Bluejay Books, Inc., 1984.

[100] V. Vinge. The coming technological singularity, 1993. VISION-21 Symposium sponsored by NASA Lewis Research Center, and Whole Earth Review, Winter issue.

[101] P. J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1, 1988.

[102] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber. Recurrent policy gradients. *Logic Journal of IGPL*, 18(2):620–634, 2010.

[103] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural evolution strategies. In *Congress of Evolutionary Computation (CEC 2008)*, 2008.

[104] R. J. Williams and D. Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Back-propagation: Theory, Architectures and Applications*. Hillsdale, NJ: Erlbaum, 1994.

[105] X. Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 4:203–222, 1993.

[106] S. Yi, F. Gomez, and J. Schmidhuber. Planning to be surprised: Optimal Bayesian exploration in dynamic environments. In *Proc. Fourth Conference on Artificial General Intelligence (AGI), Google, Mountain View, CA*, 2011.