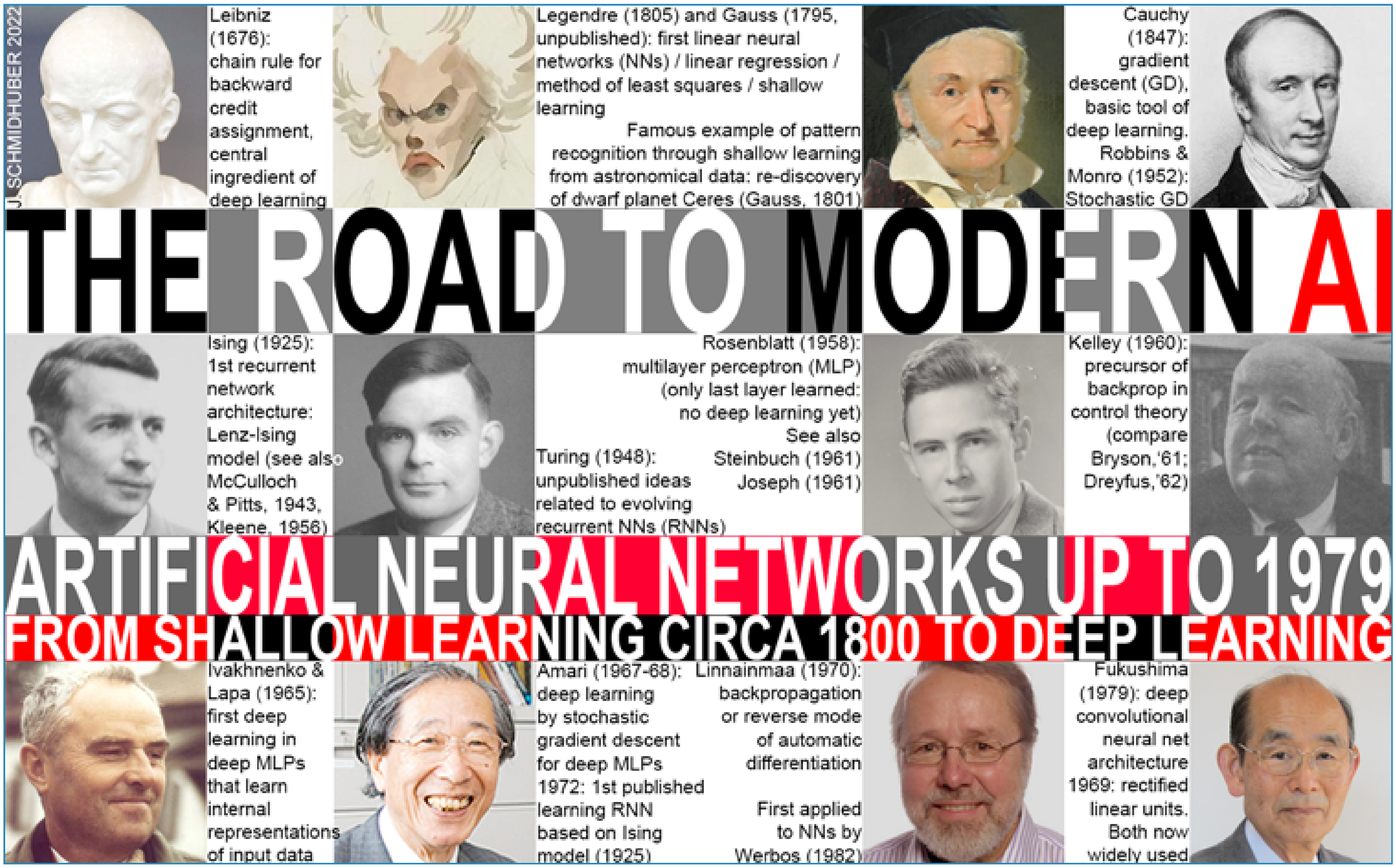


Jürgen Schmidhuber, KAUST AII, Swiss AI Lab IDSIA, USI
Pronounce: You_again Shmidhoobuh
Technical Report IDSIA-22-22 (v2), IDSIA, 12/29/2022
https://people.idsia.ch/~juergen/deep-learning-history.html

AI Blog
@SchmidhuberAI
juergen@idsia.ch
arXiv:2212.11279

Annotated History of Modern AI and Deep Learning

Abstract. Machine learning (ML) is the science of credit assignment: finding patterns in observations that predict the consequences of actions and help to improve future performance. Credit assignment is also required for human understanding of how the world works, not only for individuals navigating daily life, but also for academic professionals like historians who interpret the present in light of past events. Here I focus on the history of modern artificial intelligence (AI) which is dominated by artificial neural networks (NNs) and deep learning,[DL1-4] both conceptually closer to the old field of cybernetics than to what's been called AI since 1956 (e.g., expert systems and logic programming). A modern history of AI will emphasize breakthroughs outside of the focus of traditional AI text books, in particular, mathematical foundations of today's NNs such as the chain rule (1676), the first NNs (linear regression, circa 1800), and the first working deep learners (1965-). From the perspective of 2022, I provide a timeline of the—in hindsight—most important relevant events in the history of NNs, deep learning, AI, computer science, and mathematics in general, crediting those who laid foundations of the field. The text contains numerous hyperlinks to relevant overview sites from my AI Blog. It also debunks certain popular but misleading historic accounts of deep learning, and supplements my previous deep learning survey[DL1] which provides hundreds of additional

references. Finally, to round it off, I'll put things in a broader historic context spanning the time since the Big Bang until when the universe will be many times older than it is now. The present piece is also the draft of a chapter of my upcoming AI book.

Disclaimer. Some say a history of deep learning should not be written by someone who has helped to shape it—*"you are part of history not a historian."*[CONN21] I cannot subscribe to that point of view. Since I seem to know more about deep learning history than others,[S20][DL3,DL3a][T22][DL1-2] I consider it my duty to document and promote this knowledge, even if that seems to imply a conflict of interest, as it means prominently mentioning my own team's work, because (as of 2022) the most cited NNs are based on it.[MOST] Future AI historians may correct any era-specific potential bias.

Table of Contents

Introduction

Over time, certain historic events have become more important in the eyes of certain beholders. For example, the Big Bang of 13.8 billion years ago is now widely considered an essential moment in the history of everything. Until a few decades ago, however, it has remained completely unknown to earthlings, who for a long time have entertained quite erroneous ideas about the origins of the universe (see the final section for more on the world's history). Currently accepted histories of many more limited subjects are results of similarly radical revisions. Here I will focus on the history of artificial intelligence (AI), which also isn't quite what it used to be.

A history of AI written in the 1980s would have emphasized topics such as theorem proving,[GOD][GOD34][ZU48][NS56] logic programming, expert systems, and heuristic search.[FEI63,83][LEN83] This would be in line with topics of a 1956 conference in Dartmouth, where the term "AI" was coined by John McCarthy as a way of describing an old area of research seeing renewed interest. *Practical AI* dates back at least to 1914, when Leonardo Torres y Quevedo (see below) built the first working chess end game player[BRU1-4] (back then chess was considered as an activity restricted to the realms of intelligent creatures). AI *theory* dates back at least to 1931-34 when Kurt Gödel (see below) identified fundamental limits of any type of computation-based AI.[GOD][BIB3][GOD21,a,b]

A history of AI written in the early 2000s would have put more emphasis on topics such as support vector machines and kernel methods,[SVM1-4] Bayesian (actually Laplacian or possibly Saundersonian[STI83-85]) reasoning[BAY1-8][FI22] and other concepts of probability theory and statistics,[MM1-5][NIL98][RUS95] decision trees,[e.g.,[MIT97]] ensemble methods,[ENS1-4] swarm intelligence,[SW1] and evolutionary computation.[EVO1-7]([TUR1],unpublished) Why? Because back then such techniques drove many successful AI applications.

A history of AI written in the 2020s must emphasize concepts such as the even older chain rule[LEI07] and deep nonlinear artificial neural networks (NNs) trained by gradient descent,[GD'] in particular, feedback-based recurrent networks, which are general computers whose programs are weight matrices.[AC90] Why? Because many of the most famous and most commercial recent AI applications depend on them.[DL4]

Such NN concepts are actually conceptually close to topics of the MACY conferences (1946-1953)[MACY51] and the *1951 Paris conference on calculating machines and human thought*, now often viewed as the first conference on AI.[AI51][BRO21][BRU4] However, before 1956, much of what's now called AI was still called *cybernetics*, with a focus very much in line with modern AI based on "deep learning" with NNs.[DL1-2][DEC]

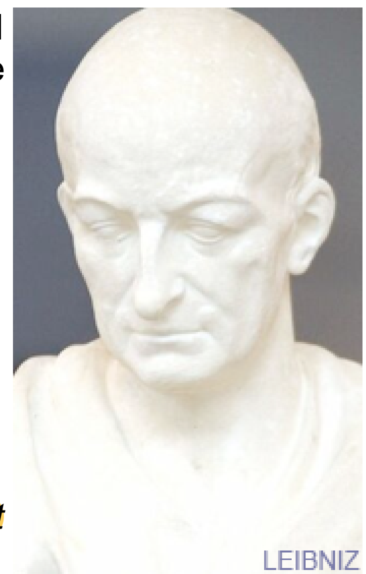
Some of the past NN research was inspired by the human brain, which has on the order of 100 billion neurons, each connected to 10,000 other neurons on average. Some are input neurons that feed the rest with data (sound, vision, tactile, pain, hunger). Others are output neurons that control muscles. Most neurons are hidden in between, where thinking takes place. Your brain apparently learns by changing the strengths or weights of the connections, which determine how strongly neurons influence each other, and which seem to encode all your lifelong experience. Similar for our *artifical* NNs, which learn better than previous methods to recognize speech or handwriting or video, minimize pain, maximize pleasure, drive cars, etc.[MIR](Sec. 0)[DL1-4]

How can NNs learn all of this? In what follows, I shall highlight essential historic contributions that made this possible. Since virtually all of the fundamental concepts of modern AI were derived in previous millennia, the section titles below emphasize developments only up to the year 2000. However, many of the sections mention the later impact of this work in the new millennium, which brought numerous improvements in hardware and software, a bit like the 20th century brought numerous improvements of the cars invented in the 19th.

The present piece also debunks a frequently repeated, misleading "history of deep learning"[S20][DL3,3a] which ignores most of the pioneering work mentioned below.[T22] See Footnote 6. The title image of the present article is a reaction to an erroneous piece of common knowledge which says[T19] that the use of NNs *"as a tool to help computers recognize patterns and simulate human intelligence had been introduced in the 1980s,"* although such NNs appeared long before the 1980s.[T22] Ensuring proper credit assignment in all of science is of great importance to me—just as it should be to all scientists—and I encourage an interested reader to also take a look at some of my letters on this in *Science* and *Nature*, e.g., on the history of aviation,[NASC1-2] the telephone,[NASC3] the computer,[NASC4-7] resilient robots,[NASC8] and scientists of the 19th century.[NASC9]

Finally, to round it off, I'll put things in a broader historic context spanning the time since the Big Bang until when the universe will be many times older than it is now.

1676: The Chain Rule For Backward Credit Assignment



In 1676, Gottfried Wilhelm Leibniz published the chain rule of differential calculus in a memoir (albeit with a sign error of all things!); Guillaume de l'Hopital described it in his 1696 textbook on Leibniz' differential calculus.[LEI07-10][L84] Today, this rule is central for credit assignment in deep neural networks (NNs). Why? The most popular NNs have nodes or neurons that compute differentiable functions of inputs from other neurons, which in turn compute differentiable functions of inputs from other neurons, and so on. The question is: how will the output of the final function change if we modify the parameters or weights of an earlier function a bit? The chain rule is the basic tool for computing the answer.

This answer is used by the technique of *gradient descent* (GD), apparently first proposed by Augustin-Louis Cauchy in 1847[GD'] (and much later by Jacques Hadamard[GD'']; the stochastic version called SGD is due to Herbert Robbins and Sutton Monro (1951)[STO51-52]). To teach an NN to translate input patterns from a training set into desired output patterns, all NN weights are iteratively changed a bit in the direction of the biggest local improvement, to create a slightly better NN, and so on, until a satisfactory solution is found.

Footnote 1. In 1684, Leibniz was also the first to publish "modern" calculus;[L84][SON18][MAD05][LEI21,a,b] later Isaac Newton was also credited for his unpublished work.[SON18] Their priority dispute,[SON18] however, did not encompass the chain rule.[LEI07-10] Of course, both were building on earlier work: in the 2nd century B.C., Archimedes (perhaps the greatest scientist ever[ARC06]) paved the way for *infinitesimals* and published special cases of calculus, e.g., for spheres and parabola segments, building on even earlier work in ancient Greece. Fundamental work on calculus was also conducted in the 14th century by Madhava of Sangamagrama and colleagues of the Indian Kerala school.[MAD86-05]

Footnote 2. Remarkably, Leibniz (1646-1714, *aka* "the world's first computer scientist"[LA14]) also laid foundations of modern computer science. He designed the first machine that could perform all four arithmetic operations (1673), and the first with an *internal memory*.[BL16] He described the principles of *binary computers* (1679)[L79][L03][LA14][HO66][LEI21,a,b] employed by virtually all modern machines. His formal *Algebra of Thought* (1686)[L86][WI48] was deductively equivalent[LE18] to the much later *Boolean Algebra* (1847).[BOO] His *Characteristica Universalis & Calculus Ratiocinator* aimed at answering all possible questions through computation;[WI48] his *"Calculemus!"* is one of the defining quotes of the age of enlightenment. It is quite remarkable that he is *also* responsible for the chain rule, foundation of "modern" deep learning, a key subfield of modern computer science.

Footnote 3. Some claim that the backpropagation algorithm (discussed further down; now widely used to train deep NNs) is just the chain rule of Leibniz (1676) & L'Hopital (1696).[CONN21] No, it is the efficient way of applying the chain rule to big networks with differentiable nodes (there are also many inefficient ways of doing this).[T22] It was not published until 1970, as discussed below.[BP1,4,5]

~1800: First NN / Linear Regression / Shallow Learning

In 1805, Adrien-Marie Legendre published what's now often called a linear neural network (NN). Later Johann Carl Friedrich Gauss was also credited for earlier unpublished work on this done circa 1795.[STI81]


LEGENDRE

This NN from over 2 centuries ago has two layers: an input layer with several input units, and an output layer. For simplicity, let's assume the latter consists of a single output unit. Each input unit can hold a real-valued number and is connected to the output by a connection with a real-valued weight. The NN's output is the sum of the products of the inputs and their weights. Given a training set of input vectors and desired target values for each of them, the NN weights are adjusted such that the sum of the squared errors between the NN outputs and the corresponding targets is minimized.

Of course, back then this was not called an NN. It was called the method of least squares, also widely known as linear regression. But it is *mathematically identical* to today's linear NNs: same basic algorithm, same error function, same adaptive parameters/weights. Such simple NNs perform "shallow learning" (as opposed to "deep learning" with many nonlinear layers). In fact, many NN courses start by introducing this method, then move on to more complex, deeper NNs.

Perhaps the first famous example of pattern recognition through shallow learning dates back over 200 years: the re-discovery of the dwarf planet Ceres in 1801 through Gauss, who had data points from previous astronomical observations, then used various tricks to adjust the parameters of a predictor, which essentially learned to generalise from the training data to correctly predict the new location of Ceres.

Footnote 4. Today, students of all technical disciplines are required to take math classes, in particular, analysis, linear algebra, and statistics. In all of these fields, essential results and methods are (at least partially) due to Gauss: the fundamental theorem of algebra, Gauss elimination, the Gaussian distribution of statistics, etc. The so-called "greatest mathematician since antiquity" also pioneered differential geometry, number theory (his favorite subject), and non-Euclidean geometry. Furthermore, he made major contributions to astronomy and physics. Modern engineering including AI would be unthinkable without his results.

Footnote 5. "Shallow learning" with NNs experienced a new wave of popularity in the late 1950s. Rosenblatt's perceptron (1958)[R58] combined a linear NN as above with an output threshold function to obtain a pattern classifier (compare his more advanced work on multi-layer networks discussed below). Joseph[R61] mentions an even earlier perceptron-like device by Farley & Clark. Widrow & Hoff's similar Adaline learned in 1962.[WID62]

1920-1925: First Recurrent Network Architecture

Like the human brain, but unlike the more limited *feedforward* NNs (FNNs), recurrent NNs (RNNs) have feedback connections, such that one can follow directed connections from certain internal nodes to others and eventually end up where one started. This is essential for implementing a memory of past events during sequence processing.

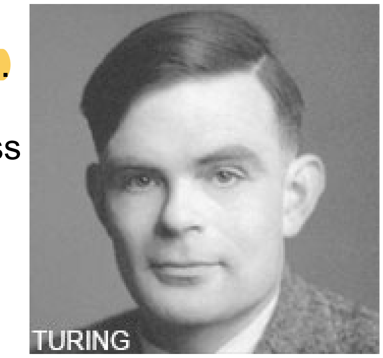
The first *non-learning* RNN architecture (the Ising model or Lenz-Ising model) was introduced and analyzed by physicists Ernst Ising and Wilhelm Lenz in the 1920s.[L20][I24,I25][K41][W45][T22] It settles into an equilibrium state in response to input conditions, and is the foundation of the first *learning* RNNs (see below).

Non-learning RNNs were also discussed in 1943 by neuroscientists Warren McCulloch und Walter Pitts[MC43] and formally analyzed in 1956 by Stephen Cole Kleene.[K56]

~1972: First Published Learning Artificial RNNs

In 1972, Shun-Ichi Amari made the Lenz-Ising recurrent architecture *adaptive* such that it could learn to associate input patterns with output patterns by changing its connection weights.[AMH1]

See also Stephen Grossberg's work on biological networks,[GRO69] David Marr's[MAR71] and Teuvo Kohonen's[KOH72] work, and Kaoru Nakano's learning RNN.[NAK72]

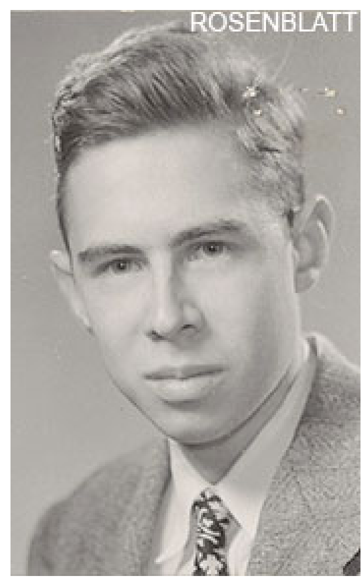


10 years later, the Amari network was republished (and its storage capacity analyzed).[AMH2] Some called it the Hopfield Network (!) or Amari-Hopfield Network.[AMH3] It does not process sequences but settles into an equilibrium in response to static input patterns. However, Amari (1972) also had a sequence-processing generalization thereof.[AMH1]

Remarkably, already in 1948, Alan Turing wrote up ideas related to artificial evolution and learning RNNs. This, however, was first published many decades later,[TUR1] which explains the obscurity of his thoughts here.[TUR21] (Margin note: it has been pointed out that the famous "Turing Test" should actually be called the "Descartes Test."[TUR3,a,b][TUR21])

Today, the most popular RNN is the Long Short-Term Memory (LSTM) mentioned below, which has become the most cited NN of the 20th century.[MOST]

1958: Multilayer Feedforward NN (without Deep Learning)



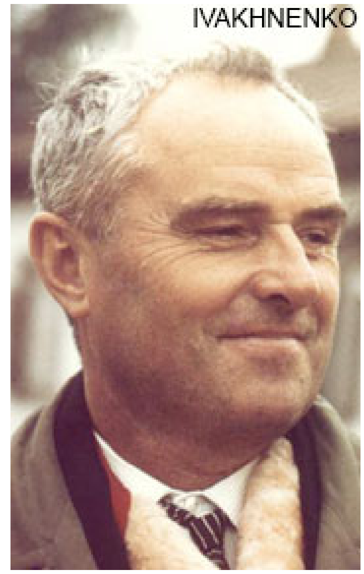
In 1958, Frank Rosenblatt not only combined linear NNs and threshold functions (see the section on shallow learning since 1800), he also had more interesting, deeper *multilayer* perceptrons (MLPs).[R58] His MLPs had a non-learning first layer with randomized weights and an adaptive output layer. Although this was not yet deep learning, because only the last layer learned,[DL1] Rosenblatt basically had what much later was rebranded as *Extreme Learning Machines (ELMs)* without proper attribution.[ELM1-2][CONN21][T22]

MLPs were also discussed in 1961 by Karl Steinbuch[ST61-95] and Roger David Joseph[R61] (1961). See also Oliver Selfridge's multilayer Pandemonium[SE59] (1959).

Rosenblatt (1962) even wrote about *"back-propagating errors"* in an MLP with a hidden layer[R62] although he did not yet have a general *deep learning* algorithm for deep MLPs. What's now called backpropagation is quite different and was first published in 1970, as discussed below.[BP1-BP5][BPA-C]

Today, the most popular FNN is a version of the LSTM-based Highway Net (mentioned below) called ResNet,[HW1-3] which has become the most cited NN of the 21st century.[MOST]

1965: First Deep Learning



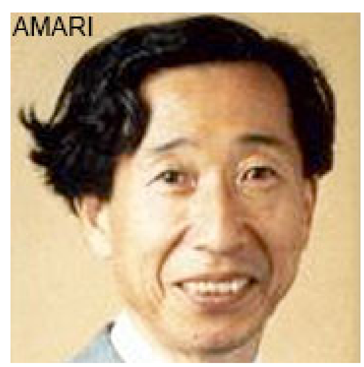
Successful learning in *deep* feedforward network architectures started in 1965 in the Ukraine (back then the USSR) when Alexey Ivakhnenko & Valentin Lapa introduced the first general, working learning algorithms for deep MLPs with arbitrarily many hidden layers (already containing the now popular multiplicative gates).[DEEP1-2][DL1-2][FDL] A paper of 1971[DEEP2] already described a deep learning net with 8 layers, trained by their highly cited method which was still popular in the new millennium,[DL2] especially in Eastern Europe, where much of Machine Learning was born.[MIR](Sec. 1)[R8]

Given a training set of input vectors with corresponding target output vectors, layers are incrementally grown and trained by regression analysis, then pruned with the help of a separate validation set, where regularisation is used to weed out superfluous units. The numbers of layers and units per layer are learned in problem-dependent fashion.

Like later deep NNs, Ivakhnenko's nets learned to create hierarchical, distributed, *internal representations* of incoming data.

He did not call them deep learning NNs, but that's what they were. In fact, the ancient term *"deep learning"* was first introduced to Machine Learning much later by Dechter (1986), and to NNs by Aizenberg et al (2000).[DL2] (Margin note: our 2005 paper on deep learning[DL6,6a] was the first machine learning publication with the word combination *"learn deep"* in the title.[T22])

1967-68: Deep Learning by Stochastic Gradient Descent



Ivakhnenko and Lapa (1965, see above) trained their deep networks layer by layer. In 1967, however, Shun-Ichi Amari suggested to train MLPs with many layers in non-incremental end-to-end fashion from scratch by stochastic gradient descent (SGD),[GD1] a method proposed in 1951 by Robbins & Monro.[STO51-52]

Amari's implementation[GD2,GD2a] (with his student Saito) learned *internal representations* in a five layer MLP with two modifiable layers, which was trained to classify non-linearly separable pattern classes. Back then compute was billions of times more expensive than today.

See also Iakov Zalmanovich Tsypkin's even earlier work on gradient descent-based on-line learning for non-linear systems.[GDa-b]

Remarkably, as mentioned above, Amari also published learning RNNs in 1972.[AMH1]

1970: Backpropagation. 1982: For NNs. 1960: Precursor.

In 1970, Seppo Linnainmaa was the first to publish what's now known as backpropagation, the famous algorithm for credit assignment in networks of differentiable nodes,[BP1,4,5] also known as "reverse mode of automatic differentiation." It is now the foundation of widely used NN software packages such as PyTorch and Google's Tensorflow.

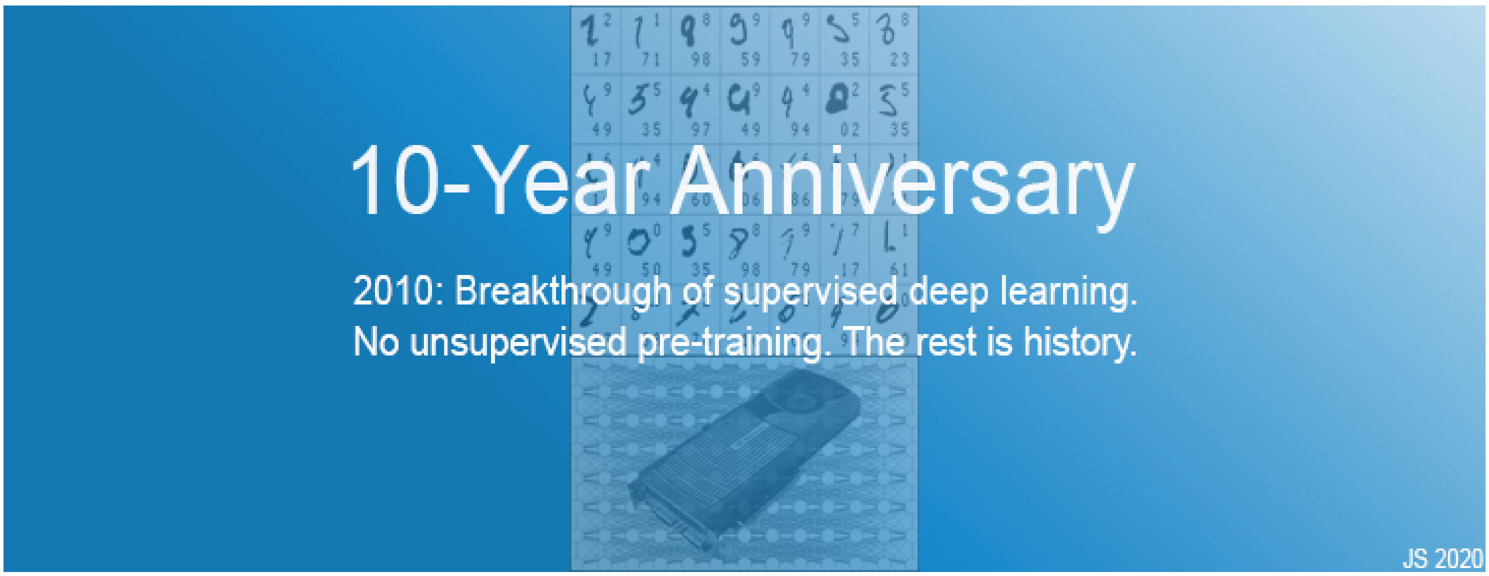
In 1982, Paul Werbos proposed to use the method to train NNs,[BP2] extending ideas in his 1974 thesis.

In 1960, Henry J. Kelley already had a precursor of backpropagation in the field of control theory;[BPA] see also later work of the early 1960s by Stuart Dreyfus and Arthur E. Bryson.[BPB][BPC][R7] Unlike Linnainmaa's general method,[BP1] the systems of the 1960s[BPA-C] backpropagated derivative information through standard Jacobian matrix calculations from one "stage" to the previous one, neither addressing direct links across several stages nor potential additional efficiency gains due to network sparsity.

Backpropagation is essentially an efficient way of implementing Leibniz's chain rule[LEI07-10] (1676) (see above) for deep networks. Cauchy's gradient descent[GD'] uses this to incrementally weaken certain NN connections and strengthen others in the course of many trials, such that the NN behaves more and more like some teacher, which could be a human, or another NN,[UN-UN2] or something else.

By 1985, compute had become about 1,000 times cheaper than in 1970, and the first desktop computers had just become accessible in wealthier academic labs. An experimental analysis of the known method[BP1-2] by David E. Rumelhart et al. then demonstrated that backpropagation can yield useful internal representations in hidden layers of NNs.[RUM] At least for supervised learning, backpropagation is generally more efficient than Amari's above-mentioned deep learning through the more general SGD method (1967), which learned useful internal representations in NNs about 2 decades earlier.[GD1-2a]

It took 4 decades until the backpropagation method of 1970[BP1-2] got widely accepted as a training method for *deep* NNs. Before 2010, many thought that the training of NNs with many layers requires unsupervised pre-training, a methodology introduced by myself in 1991[UN][UN0-3] (see below), and later championed by others (2006).[UN4] In fact, it was claimed[VID1] that "nobody in their right mind would ever suggest" to apply plain backpropagation to deep NNs. However, in 2010, our team with my outstanding Romanian postdoc Dan Ciresan[MLP1-2] showed that deep FNNs can be trained by plain backpropagation and do not at all require unsupervised pre-training for important applications.[MLP2]



Our system set a new performance record[MLP1] on the back then famous and widely used image recognition benchmark called MNIST. This was achieved by greatly accelerating deep FNNs on highly parallel graphics processing units called GPUs (as first done for shallow NNs with few layers by Jung & Oh in 2004[GPUNN]). A reviewer called this a "wake-up call to the machine learning community." Today, everybody in the field is pursuing this approach.

Footnote 6. Unfortunately, several authors who re-published backpropagation in the 1980s did not cite the prior art—not even in later surveys.[T22] In fact, as mentioned in the introduction, there is a broader, frequently repeated, misleading "history of deep learning"[S20] which ignores most of the pioneering work mentioned in the previous sections.[T22][DLC] This "alternative history" essentially goes like this: *"In 1969, Minsky & Papert[M69] showed that shallow NNs without hidden layers are very limited and the field was abandoned until a new generation of neural network researchers took a fresh look at the problem in the 1980s."*[S20] However, the 1969 book[M69] addressed a "problem" of Gauss & Legendre's shallow learning (circa 1800)[DL1-2] that had already been solved 4 years prior by Ivakhnenko & Lapa's popular deep learning method,[DEEP1-2][DL2] and then also by Amari's SGD for MLPs.[GD1-2] Minsky neither cited this work nor corrected his book later.[HIN](Sec. I)[T22] And even recent papers promulgate this revisionist narrative of deep learning, apparently to glorify later contributions of their authors (such as the Boltzmann machine[BM][HIN][SK75][G63][T22]) without relating them to the original work,[DLC][S20][T22] although the true history is well-known. Deep learning research was alive and kicking in the

1960s-70s, especially outside of the Anglosphere.[DEEP1-2][GD1-3][CNN1][DL1-2][T22] Blatant misattribution and unintentional[PLAG1][CONN21] or intentional[FAKE2] plagiarism are still tainting the entire field of deep learning.[T22] Scientific journals "need to make clearer and firmer commitments to self-correction,"[SV20] as is already the standard in other scientific fields.

1979: First Deep Convolutional NN (1969: ReLUs)

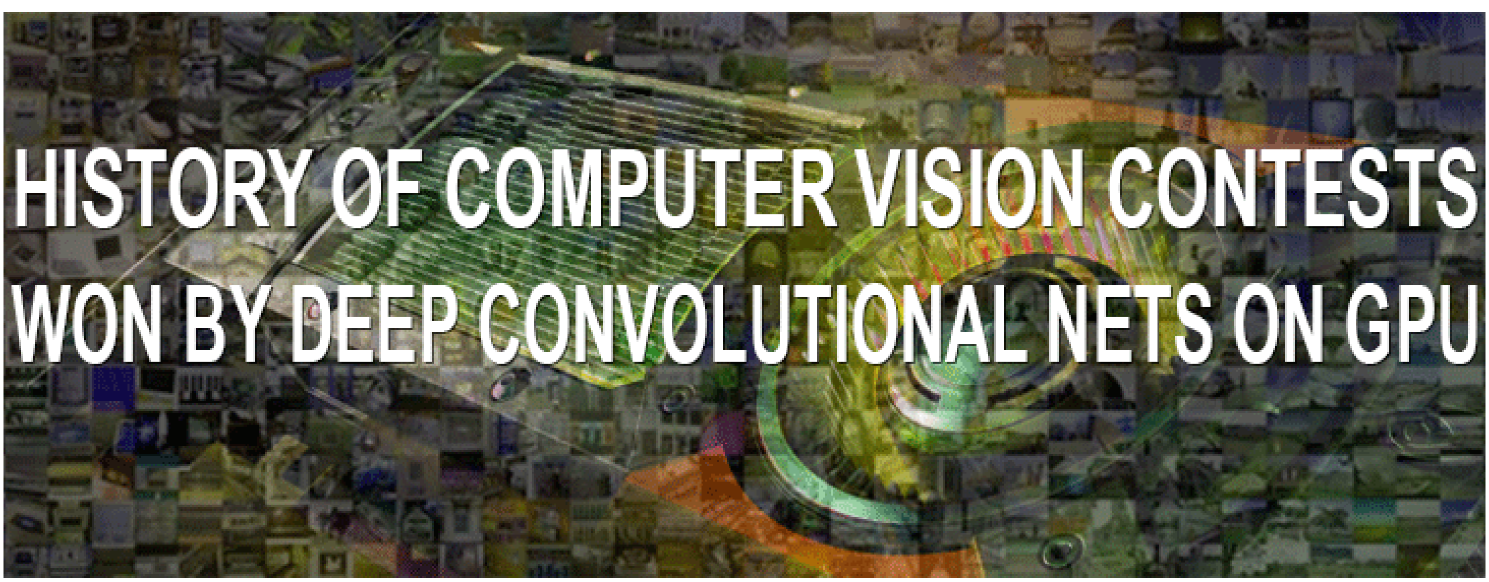
Computer Vision was revolutionized in the 2010s by a particular feedforward NN called the convolutional NN (CNN).[CNN1-4] The basic CNN architecture with alternating convolutional and downsampling layers is due to Kunihiko Fukushima (1979). He called it Neocognitron.[CNN1]

Remarkably, already 10 years earlier, Fukushima also introduced *rectified linear units* (ReLUs) for NNs (1969).[RELU1] They are now widely used in CNNs and other NNs.

In 1987, NNs with convolutions were combined by Alex Waibel with weight sharing and backpropagation (see above),[BP1-2] and applied to speech.[CNN1a] Waibel did not call this CNNs but TDNNs.

A popular downsampling variant called max-pooling was introduced by Yamaguchi et al. for TDNNs in 1990[CNN3a] and by Juan Weng et al. for higher-dimensional CNNs in 1993.[CNN3]

Since 1989, Yann LeCun's team has contributed improvements of CNNs, especially for images.[CNN2,4][T22] Baldi and Chauvin (1993) had the first application of CNNs with backpropagation to biomedical/biometric images.[BA93]



CNNs became more popular in the ML community much later in 2011 when my own team greatly sped up the training of deep CNNs (Dan Ciresan et al., 2011).[GPUCNN1,3,5] Our fast GPU-based[GPUNN][GPUCNN5] CNN of 2011[GPUCNN1] known as DanNet[DAN,DAN1][R6] was a practical breakthrough, much deeper and faster than earlier GPU-accelerated CNNs of 2006.[GPUCNN] In 2011, DanNet became the first pure deep CNN to win computer vision contests.[GPUCNN2-3,5]

Competition[GPUCNN5]	Date/Deadline	Image size	Improvement	Winner
ICDAR 2011 Chinese handwriting	May 15, 2011	variable	3.8% / 28.9%	DanNet[GPUCNN1-3]
IJCNN 2011 traffic signs	**Aug 06, 2011**	variable	**68.0%** (superhuman)	**DanNet**[DAN,DAN1][R6]
ISBI 2012 image segmentation	Mar 01, 2012	512x512	26.1%	DanNet[GPUCNN3a]
ICPR 2012 medical imaging	Sep 10, 2012	2048x2048x3	8.9%	DanNet[GPUCNN8]
ImageNet 2012	Sep 30, 2012	256x256x3	41.4%	AlexNet[GPUCNN4]
MICCAI 2013 Grand Challenge	Sep 08, 2013	2048x2048x3	26.5%	DanNet[GPUCNN8]
ImageNet 2014	Aug 18, 2014	256x256x3		VGG Net[GPUCNN9]
ImageNet 2015	Sep 30, 2015	256x256x3	15.8%	ResNet,[HW2] a Highway Net[HW1] with open gates

For a while, DanNet enjoyed a monopoly. From 2011 to 2012 it won every contest it entered, winning four of them in a row (15 May 2011, 6 Aug 2011, 1 Mar 2012, 10 Sep 2012).[GPUCNN5] In particular, at IJCNN 2011 in Silicon Valley, DanNet blew away the competition and achieved the first superhuman visual pattern recognition[DAN1] in an international contest. DanNet was also the first deep CNN to win: a Chinese handwriting contest (ICDAR 2011), an image segmentation contest (ISBI, May 2012), a contest on object detection in large images (ICPR, 10 Sept 2012), and— at the same time—a medical imaging contest on cancer detection.[GPUCNN8] In 2010, we introduced DanNet to Arcelor Mittal, the world's largest steel producer, and were able to greatly improve steel defect detection.[ST] To the best of my knowledge, this was the first deep learning breakthrough in heavy industry. In July 2012, our CVPR paper on DanNet[GPUCNN3] hit the computer vision community. 5 months later, the similar GPU-accelerated AlexNet won the ImageNet[IM09] 2012 contest.[GPUCNN4-5][R6] Our CNN image scanners were 1000 times faster than previous methods.[SCAN] This attracted tremendous interest from the healthcare industry. Today IBM, Siemens, Google and many startups are pursuing this approach. The VGG network (ImageNet 2014 winner)[GPUCNN9] and other highly cited CNNs[RCNN1-3] further extended the DanNet of 2011.[MIR](Sec. 19)[MOST]

AlexNet —> VGG —> ResNet

ResNet, the ImageNet 2015 winner[HW2] (Dec 2015) and currently the most cited NN,[MOST] is a version (with open gates) of our earlier Highway Net (May 2015).[HW1-3][R5] The Highway Net (see below) is actually the feedforward net version of our vanilla LSTM (see below).[LSTM2] It was the first working, really deep feedforward NN with hundreds of layers (previous NNs had at most a few tens of layers).

1980s-90s: Graph NNs / Stochastic Delta Rule (Dropout) /...

NNs with rapidly changing "fast weights" were introduced by v.d. Malsburg (1981) and others.[FAST,a,b] Deep learning architectures that can manipulate structured data such as graphs[T22] were proposed in 1987 by Pollack[PO87-90] and extended/improved by Sperduti, Goller, and Küchler in the early 1990s.[SP93-97][GOL][KU][T22] See also our graph NN-like, Transformer-like Fast Weight Programmers of 1991[FWP0-1][FWP6][FWP] which learn to continually rewrite mappings from inputs to outputs (addressed below), and the work of Baldi and colleagues.[BA96-03] Today, graph NNs are used in numerous applications.

Werbos,[BP2][BPTT1] Williams,[BPTT2][CUB0-2] and others[ROB87][BPTT3][DL1] analyzed ways of implementing gradient descent[GD'][STO51-52][GDa-b][GD1-2a] in RNNs. Kohonen's self-organising maps became popular.[KOH82-89] I did work on this :)

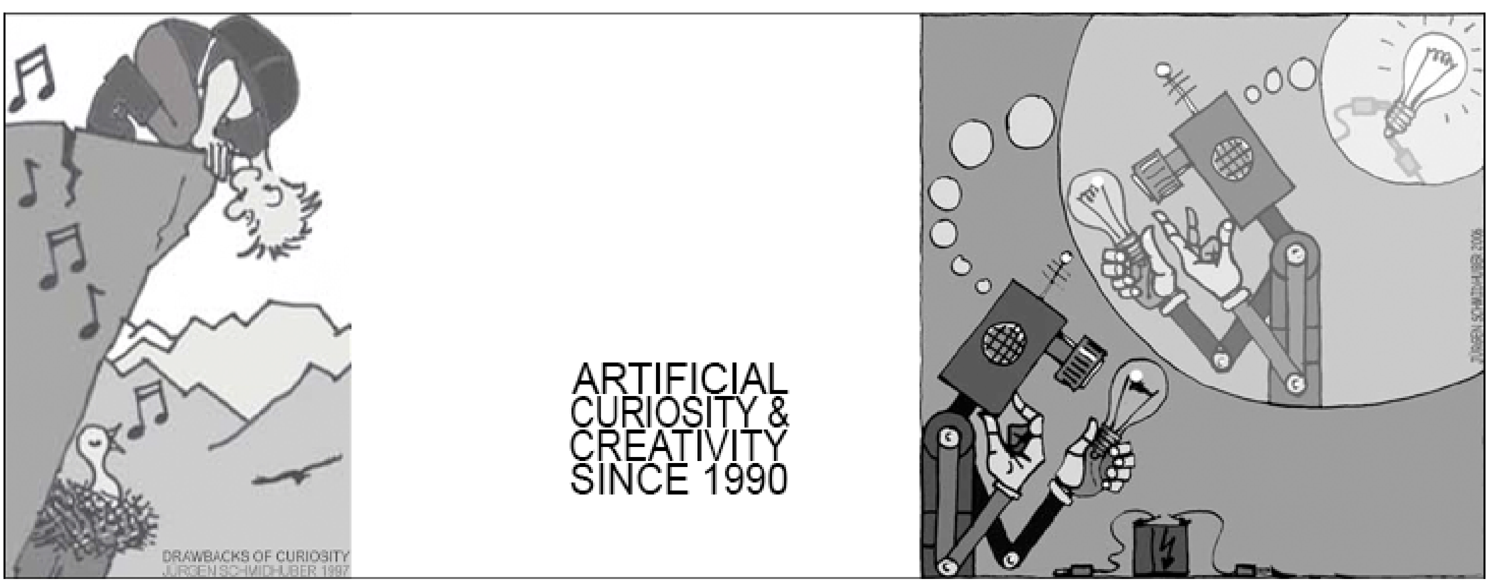
The 80s and 90s also saw various proposals of biologically more plausible deep learning algorithms that—unlike backpropagation—are local in space and time.[BB2][NAN1-4][NHE][HEL] See overviews[MIR](Sec. 15, Sec. 17) and recent renewed interest in such methods.[NAN5][FWPMETA6][HIN22]

In 1990, Hanson introduced the Stochastic Delta Rule, a stochastic way of training NNs by backpropagation. Decades later, a version of this became popular under the moniker "dropout."[Drop1-4][GPUCNN4]

Many additional papers on NNs (including RNNs) were published in the 1980s and 90s—see the numerous references in the 2015 survey.[DL1] Here, however, we mostly limit ourselves to the—in hindsight—most essential ones, given the present (ephemeral?) perspective of 2022.

Feb 1990: Generative Adversarial Networks / Curiosity

Generative Adversarial Networks (GANs) have become very popular.[MOST] They were first published in 1990 in Munich under the moniker Artificial Curiosity.[AC90-20][GAN1] Two dueling NNs (a probabilistic generator and a predictor) are trying to maximize each other's loss in a minimax game.[AC](Sec. 1) The generator (called the controller) generates probabilistic outputs (using stochastic units[AC90] like in the much later StyleGANs[GAN2]). The predictor (called the world model) sees the outputs of the controller and predicts environmental reactions to them. Using gradient descent, the predictor NN *minimizes* its error, while the generator NN tries to make outputs that *maximize* this error: one net's loss is the other net's gain.[AC90] (The world model can also be used for continual **online action planning**.[AC90][PLAN2-3][PLAN])



4 years before a 2014 paper on GANs,[GAN1] my well-known 2010 survey[AC10] summarised the generative adversarial NNs of 1990 as follows: a *"neural network as a predictive world model is used to maximize the controller's intrinsic reward, which is proportional to the model's prediction errors"* (which are minimized).

The 2014 GANs are an instance of this where the trials are very short (like in bandit problems) and the environment simply returns 1 or 0 depending on whether the controller's (or generator's) output is in a given set.[AC20][AC][T22](Sec. XVII)

Other early adversarial machine learning settings[S59][H90] were very different—they neither involved unsupervised NNs nor were about modeling data nor used gradient descent.[AC20]

The 1990 principle has been widely used for exploration in Reinforcement Learning[SIN5][OUD13][PAT17][BUR18] and for synthesis of realistic images,[GAN1,2] although the latter domain was recently taken over by Rombach et al.'s *Latent Diffusion*, another method published in Munich,[DIF1] building on Jarzynski's earlier work in physics from the previous millennium[DIF2] and more recent papers.[DIF3-5]

In 1991, I published yet another ML method based on two adversarial NNs called Predictability Minimization for creating disentangled representations of partially redundant data, applied to images in 1996.[PM0-2][AC20][R2][MIR](Sec. 7)

April 1990: NNs Generate Subgoals / Work on Command

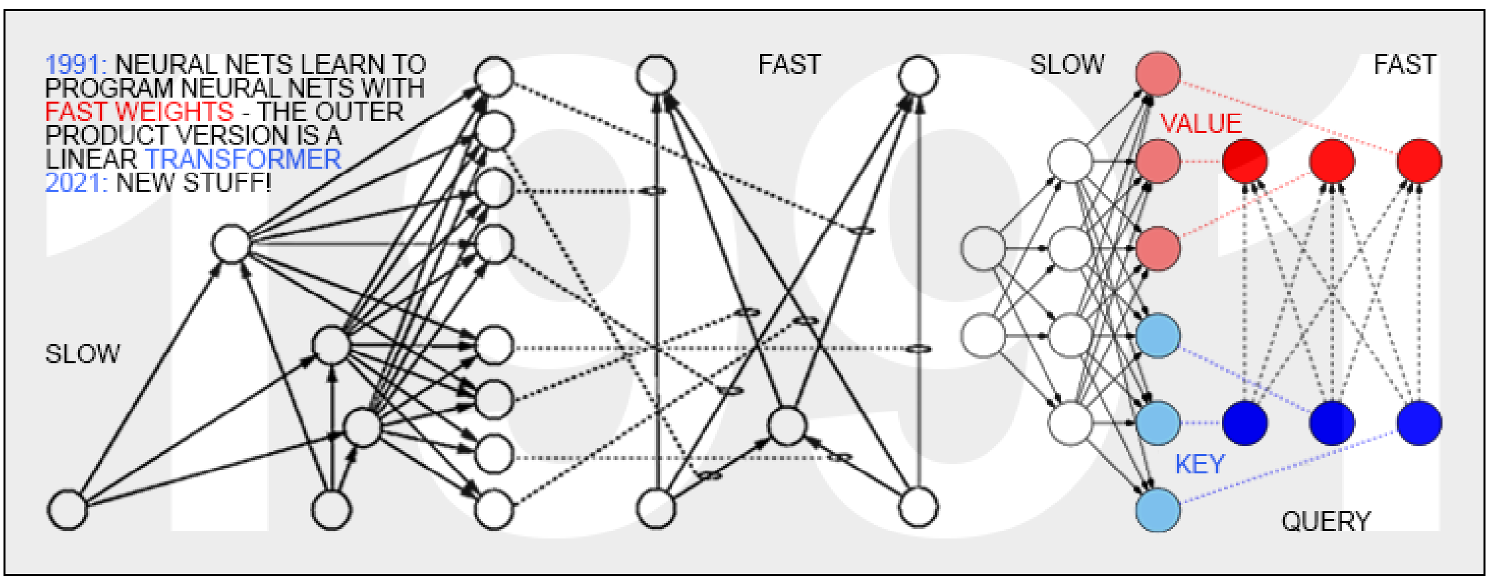
Most NNs of recent centuries were dedicated to simple pattern recognition, not to high-level reasoning, which is now considered a remaining grand challenge.[LEC] The early 1990s, however, saw first exceptions: NNs that learn to *decompose* complex spatio-temporal observation sequences into compact but meaningful *chunks*[UN0-3] (see further below), and NN-based planners of hierarchical action sequences for *compositional learning*,[HRL0] as discussed next. This work injected concepts of traditional *"symbolic"* hierarchical AI[NS59][FU77] into end-to-end differentiable *"sub-symbolic"* NNs.

In 1990, our NNs learned to generate hierarchical action plans with end-to-end differentiable NN-based subgoal generators for Hierarchical Reinforcement Learning (HRL).[HRL0] Soon afterwards, this was also done with recurrent NNs that learn to generate sequences of subgoals.[HRL1-2][PHD][MIR](Sec. 10) An RL machine gets extra *command inputs* of the form *(start, goal)*. An evaluator NN learns to predict the current rewards/costs of going from *start* to *goal*. An (R)NN-based subgoal generator also sees *(start, goal)*, and uses (copies of) the evaluator NN to learn by gradient descent a sequence of cost-minimising intermediate subgoals. The RL machine tries to use such subgoal sequences to achieve final goals. The system is learning action plans at multiple levels of abstraction and multiple time scales and solves (at least in principle) what recently (2022) has been called an "open problem."[LEC]

Compare other NNs that have "worked on command" since April 1990, in particular, for learning selective attention,[ATT0-3] artificial curiosity and self-invented problems,[PP][PPa,1,2][AC] upside-down reinforcement learning[UDRL1-2] and its generalizations.[GGP]

March 1991: Transformers with Linearized Self-Attention

Recently, Transformers[TR1] have been all the rage, e.g., generating human-sounding texts.[GPT3] Transformers with "linearized self-attention"[TR5-6] were first published in March 1991[FWP0-1][FWP6][FWP] (apart from normalisation—see tweet of 2022 for 30-year anniversary). These so-called "Fast Weight Programmers" or "Fast Weight Controllers"[FWP0-1] separated storage and control like in traditional computers, but in an end-to-end-differentiable, adaptive, fully neural way (rather than in a hybrid fashion[PDA1-2][DNC]). The "self-attention" in standard Transformers[TR1-4] combines this with a projection and *softmax* (using attention terminology like the one I introduced in 1993[ATT][FWP2][R4]).



Today's Transformers heavily use unsupervised pre-training[UN0-3] (see next section), another deep learning methodology first published in our Annus Mirabilis of 1990-1991.[MIR][MOST]

The 1991 fast weight programmers also led to meta-learning self-referential NNs that can run their own weight change algorithm or learning algorithm on themselves, and improve it, and improve the way they improve it, and so on. This work since 1992[FWPMETA1-9][HO1] extended my 1987 diploma thesis,[META1] which introduced algorithms not just for learning but also for meta-learning or learning to learn,[META] to learn better learning algorithms through experience. This became very popular in the 2010s[DEC] when computers were a million times faster.

April 1991: Deep Learning by Self-Supervised Pre-Training

Today's most powerful NNs tend to be very deep, that is, they have many layers of neurons or many subsequent computational stages.[MIR] Before the 1990s, however, gradient-based training did not work well for deep NNs, only for shallow ones[DL1-2] (but see a 1989 paper[MOZ]). This *Deep Learning Problem* was most obvious for *recurrent* NNs. Like the human brain, but unlike the more limited *feedforward* NNs (FNNs), RNNs have feedback connections. This makes RNNs powerful, general purpose, parallel-sequential computers that can process input sequences of arbitrary length (think of speech data or videos). RNNs can in principle implement any program that can run on your laptop or any other computer in existence. If we want to build an *Artificial General Intelligence* (AGI), then its underlying computational substrate must be something more like an RNN than an FNN as FNNs are fundamentally insufficient; RNNs and similar systems are to FNNs as general computers are to pocket calculators. In particular, unlike FNNs, RNNs can in principle deal with problems of arbitrary depth.[DL1] Before the 1990s, however, RNNs failed to learn deep problems in practice.[MIR](Sec. 0)

To overcome this drawback through RNN-based *"general deep learning,"* I built a self-supervised RNN hierarchy that learns representations at multiple levels of abstraction and multiple self-organizing time scales:[LEC] the *Neural Sequence Chunker*[UN0] or Neural History Compressor.[UN1] Each RNN tries to solve the *pretext task* of predicting its next input, sending only unexpected inputs (and therefore also targets) to the next RNN above. The resulting compressed sequence representations greatly facilitate downstream *supervised* deep learning such as sequence classification.

Although computers back then were about a million times slower per dollar than today, by 1993, the Neural History Compressor above was able to solve previously unsolvable "very deep learning" tasks of depth > 1000[UN2] (requiring more than 1,000 subsequent computational stages—the more such stages, the deeper the learning). In 1993, we also published a *continuous* version of the Neural History Compressor.[UN3] (See also recent work on unsupervised NN-based abstraction.[OBJ1-5])

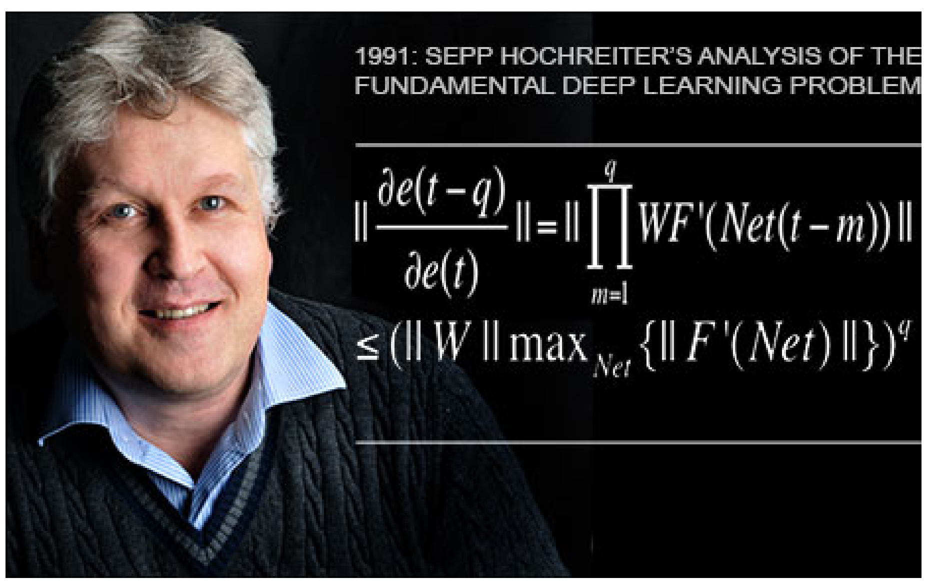
More than a decade after this work,[UN1] a similar unsupervised method for more limited *feedforward* NNs (FNNs) was published, facilitating supervised learning by unsupervised pre-training of stacks of FNNs called *Deep Belief Networks* (DBNs).[UN4] The 2006 justification was essentially the one I used in the early 1990s for my RNN stack: each higher level tries to reduce the description length (or negative log probability) of the data representation in the level below.[HIN][T22][MIR]

April 1991: Distilling one NN into another NN

The hierarchical internal representations of the neural history compressor above can be collapsed into a single recurrent NN (RNN), using my NN distillation procedure of 1991.[UN0-1][MIR] Here the knowledge of a teacher NN is "distilled" into a student NN, by training the student NN to imitate the behavior of the teacher NN (while also re-training the student NN on previously learned skills such that it does not forget them). NN distillation was also republished many years later,[DIST2][MIR][HIN][T22] and is widely used today.

Today, unsupervised pre-training is heavily used by *Transformers*[TR1-6] for natural language processing and other domains. Remarkably, Transformers with linearized self-attention were also first published[FWP0-6] in our Annus Mirabilis of 1990-1991,[MIR][MOST] together with unsupervised/self-supervised pre-training for deep learning.[UN0-3] See the previous section.

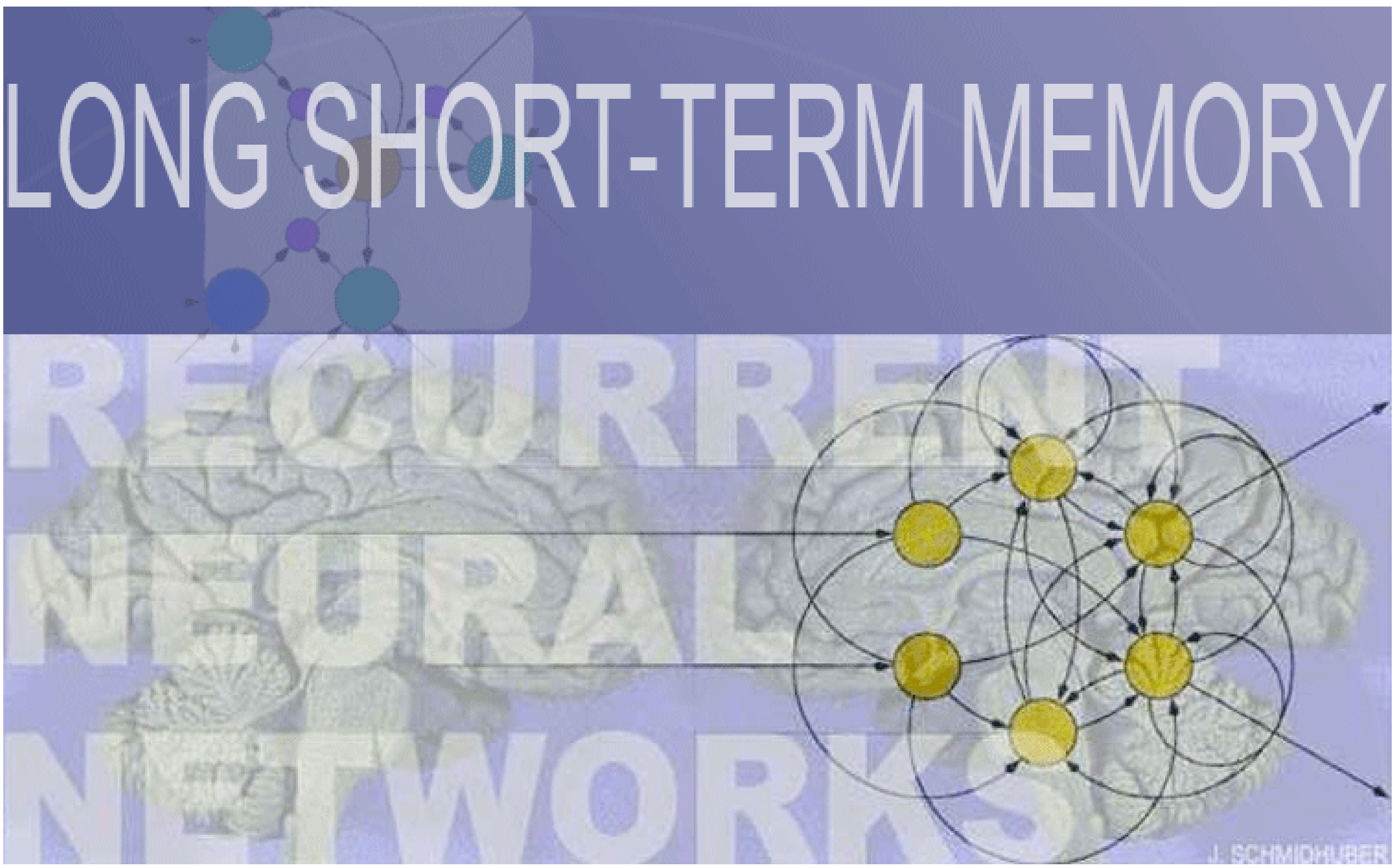
June 1991: Fundamental Problem: Vanishing Gradients



Deep learning is hard because of the Fundamental Deep Learning Problem identified and analyzed in 1991 by my first student Sepp Hochreiter in his diploma thesis which I had the pleasure to supervise.[VAN1] First he implemented the Neural History Compressor above but then did much more: he showed that deep NNs suffer from the now famous problem of vanishing or exploding gradients: in typical deep or recurrent networks, back-propagated error signals either shrink rapidly, or grow out of bounds. In both cases, learning fails (compare[VAN2]). This analysis led to basic principles of what's now called LSTM (see below).

June 1991: Roots of LSTM / Highway Nets / ResNets

The Long Short-Term Memory (LSTM) recurrent neural network[LSTM1-6] overcomes the Fundamental Deep Learning Problem identified by Sepp in his above-mentioned 1991 diploma thesis,[VAN1] which I consider one of the most important documents in the history of machine learning. It also provided essential insights for overcoming the problem, through basic principles (such as *constant error flow*) of what we called LSTM in a tech report of 1995.[LSTM0] After the main peer-reviewed publication in 1997[LSTM1][25y97] (now the most cited NN article of the 20th century[MOST]), LSTM and its training procedures were further improved on my Swiss LSTM grants at IDSIA through the work of my later students Felix Gers, Alex Graves, and others. A milestone was the *"vanilla LSTM architecture"* with forget gate[LSTM2]—the LSTM variant of 1999-2000 that everybody is using today, e.g., in Google's Tensorflow. Alex was lead author of our first successful application of LSTM to speech (2004).[LSTM10] 2005 saw the first publication of LSTM with full backpropagation through time and of bi-directional LSTM[LSTM3] (now widely used).



Another milestone of 2006 was the training method *"Connectionist Temporal Classification"* or CTC[CTC] for simultaneous alignment and recognition of sequences. Our team successfully applied CTC-trained LSTM to speech in 2007[LSTM4] (also with hierarchical LSTM stacks[LSTM14]). This led to the first superior end-to-end neural speech recognition. It was very different from hybrid methods since the late 1980s which combined NNs and traditional approaches such as Hidden Markov Models (HMMs).[BW][BRI][BOU][HYB12][T22] In 2009, through the efforts of Alex, LSTM trained by CTC became the first RNN to win international competitions, namely, three ICDAR 2009 Connected Handwriting Competitions (French, Farsi, Arabic). This attracted enormous interest from industry. LSTM was soon used for everything that involves sequential data such as speech[LSTM10-11][LSTM4][DL1] and videos. In 2015, the CTC-LSTM combination dramatically improved Google's speech recognition on the Android smartphones.[GSR15] Many other companies adopted this.[DL4] Google's new on-device speech recognition of 2019 (now on your phone, not on the server) is still based on LSTM.

1995: Neural Probabilistic Language Model

The first superior end-to-end neural machine translation was also based on LSTM. In 1995, we already had an excellent neural probabilistic text model[SNT] whose basic concepts were reused in 2003[NPM][T22]—see also Pollack's earlier work on embeddings of words and other structures[PO87][PO90] as well as Nakamura and Shikano's 1989 word category prediction model.[NPMa] In 2001, we showed that LSTM can learn languages unlearnable by traditional models such as HMMs,[LSTM13] i.e., a neural "subsymbolic" model suddenly excelled at learning "symbolic" tasks. Compute still had to get 1000 times cheaper, but by 2016, Google Translate[GT16]—whose whitepaper[WU] mentions LSTM over 50 times—was based on two connected LSTMs,[S2S] one for incoming texts, and one for outgoing translations—much better than what existed before.[DL4] By 2017, LSTM also powered Facebook's machine translation (over 30 billion translations per week—the most popular youtube video needed years to achieve only 10 billion clicks),[FB17][DL4] Apple's Quicktype on roughly 1 billion iPhones,[DL4] the voice of Amazon's Alexa,[DL4] Google's image caption generation[DL4] & automatic email answering[DL4] etc. Business Week called LSTM "arguably the most commercial AI achievement."[AV1] By 2016, more than a quarter of the awesome computational power for inference in Google's datacenters was used for LSTM (and 5% for another popular Deep Learning technique called CNNs—see above).[JOU17] And of course, our LSTM is also massively used in healthcare and medical diagnosis—a simple Google Scholar search turns up innumerable medical articles that have "LSTM" in their title.[DEC]

Through the work of my students Rupesh Kumar Srivastava and Klaus Greff, the LSTM principle also led to our Highway Network[HW1] of May 2015, the first working very deep FNN with hundreds of layers (previous NNs had at most a few tens of layers). Microsoft's

ResNet[HW2] (which won the ImageNet 2015 contest) is a version thereof (ResNets are Highway Nets whose gates are always open). The earlier Highway Nets perform roughly as well as their ResNet versions on ImageNet.[HW3] Variants of highway gates are also used for certain algorithmic tasks where the pure residual layers do not work as well.[NDR]

The LSTM / Highway Net Principle is the Core of Modern Deep Learning

Deep learning is all about NN depth.[DL1] In the 1990s, LSTMs brought essentially *unlimited* depth to supervised *recurrent* NNs; in the 2000s, the LSTM-inspired Highway Nets brought it to *feedforward* NNs. LSTM has become the most cited NN of the 20th century; the Highway Net version called ResNet the most cited NN of the 21st.[MOST] (Citations, however, are a highly questionable measure of true impact.[NAT1])

1980s-: NNs for Learning to Act Without a Teacher

The previous sections have mostly focused on deep learning for passive pattern recognition/classification. However, NNs are also relevant for Reinforcement Learning (RL), [KAE96][BER96][TD3][UNI][GM3][LSTMPG] the most general type of learning. General RL agents must discover, without the aid of a teacher, how to interact with a dynamic, initially unknown, partially observable environment in order to maximize their expected cumulative reward signals.[DL1] There may be arbitrary, a priori unknown delays between actions and perceivable consequences. The RL problem is as hard as any problem of computer science, since any task with a computable description can be formulated in the general RL framework.[UNI]

Certain RL problems can be addressed through non-neural techniques invented long before the 1980s: Monte Carlo (tree) search (MC, 1949),[MOC1-5] dynamic programming (DP, 1953),[BEL53] artificial evolution (1954),[EVO1-7]([TUR1],unpublished) alpha-beta-pruning (1959),[S59] control theory and system identification (1950s),[KAL59][GLA85] stochastic gradient descent (SGD, 1951),[STO51-52] and universal search techniques (1973).[AIT7]

Deep FNNs and RNNs, however, are useful tools for *improving* certain types of RL. In the 1980s, concepts of function approximation and NNs were combined with system identification, [WER87-89][MUN87][NGU89] DP and its online variant called Temporal Differences (TD),[TD1-3] artificial evolution,[EVONN1-3] and policy gradients.[GD1][PG1-3] Many additional references on this can be found in Sec. 6 of the 2015 survey.[DL1]

When there is a Markovian interface[PLAN3] to the environment such that the current input to the RL machine conveys all the information required to determine a next optimal action, RL with DP/TD/MC-based FNNs can be very successful, as shown in 1994[TD2] (master-level backgammon player) and the 2010s[DM1-2a] (superhuman players for Go, chess, and other games).

For more complex cases without Markovian interfaces, where the learning machine must consider not only the present input, but also the history of previous inputs, our combinations of RL algorithms and LSTM[LSTM-RL][RPG] have become standard, in particular, our LSTM trained by policy gradients (2007).[RPG07][RPG][LSTMPG]

For example, in 2018, a PG-trained LSTM was the core of OpenAI's famous Dactyl which learned to control a dextrous robot hand without a teacher.[OAI1][OAI1a] Similar for video games: in 2019, DeepMind (co-founded by a student from my lab) famously beat a pro player in the game of Starcraft, which is theoretically harder than Chess or Go[DM2] in many ways, using Alphastar whose brain has a deep LSTM core trained by PG.[DM3] An RL LSTM (with 84% of the model's total parameter count) also was the core of the famous OpenAI Five which learned to defeat human experts in the Dota 2 video game (2018).[OAI2] Bill Gates called this a *"huge milestone in advancing artificial intelligence"*.[OAI2a][MIR](Sec. 4)[LSTMPG]

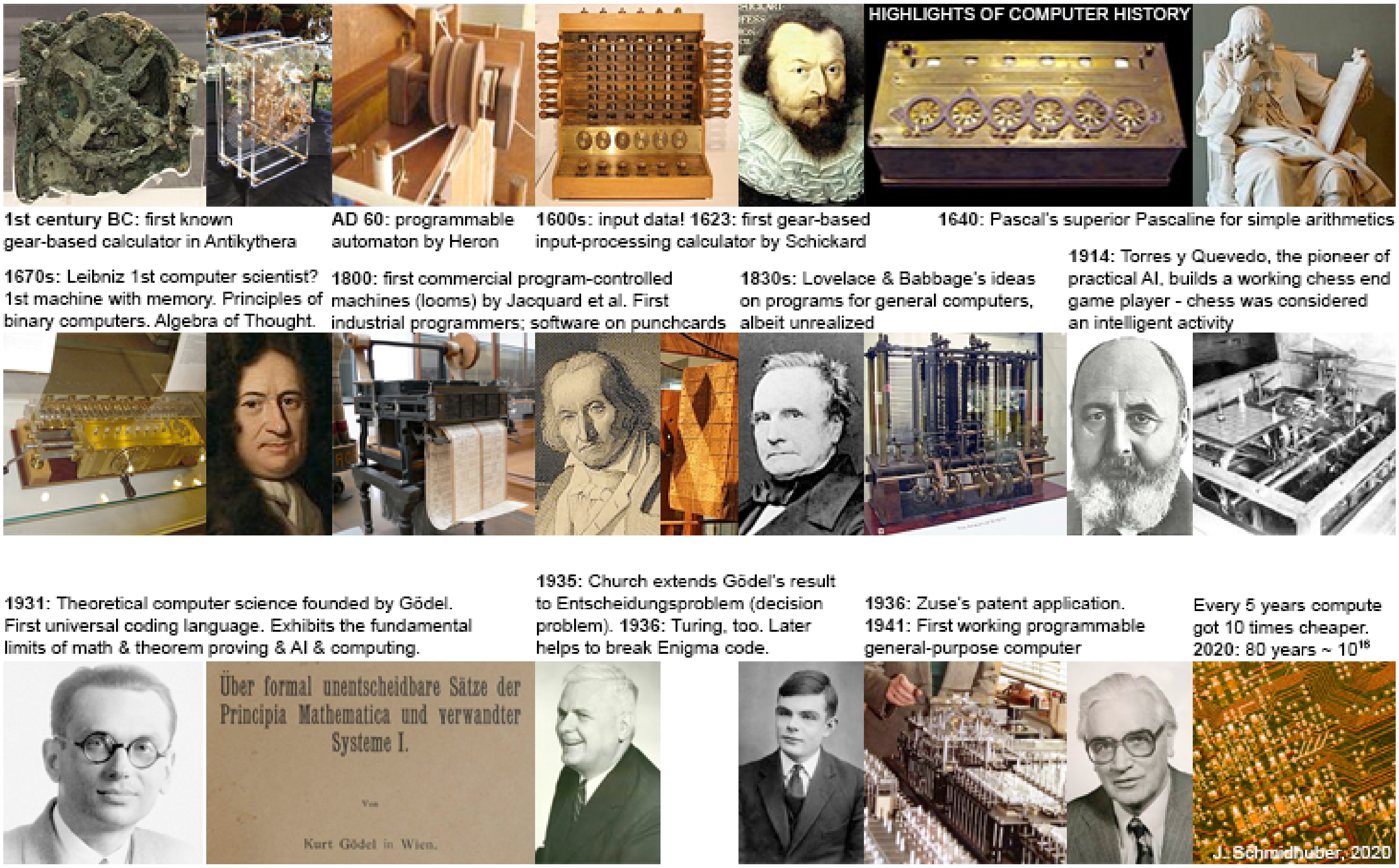
The future of RL will be about learning/composing/planning with compact spatio-temporal abstractions of complex input streams—about commonsense reasoning[MAR15] and *learning to think*.[PLAN4-5] How can NNs learn to represent percepts and action plans in a hierarchical manner, at multiple levels of abstraction, and multiple time scales?[LEC] We published answers to these questions in 1990-91: self-supervised neural history compressors[UN][UN0-3] learn to represent percepts at multiple levels of abstraction and multiple time scales (see above), while end-to-end differentiable NN-based subgoal generators[HRL3][MIR](Sec. 10) learn hierarchical action plans through gradient descent (see above). More sophisticated ways of learning to think in abstract ways were published in 1997[AC97][AC99][AC02] and 2015-18.[PLAN4-5]

It's the Hardware, Stupid!

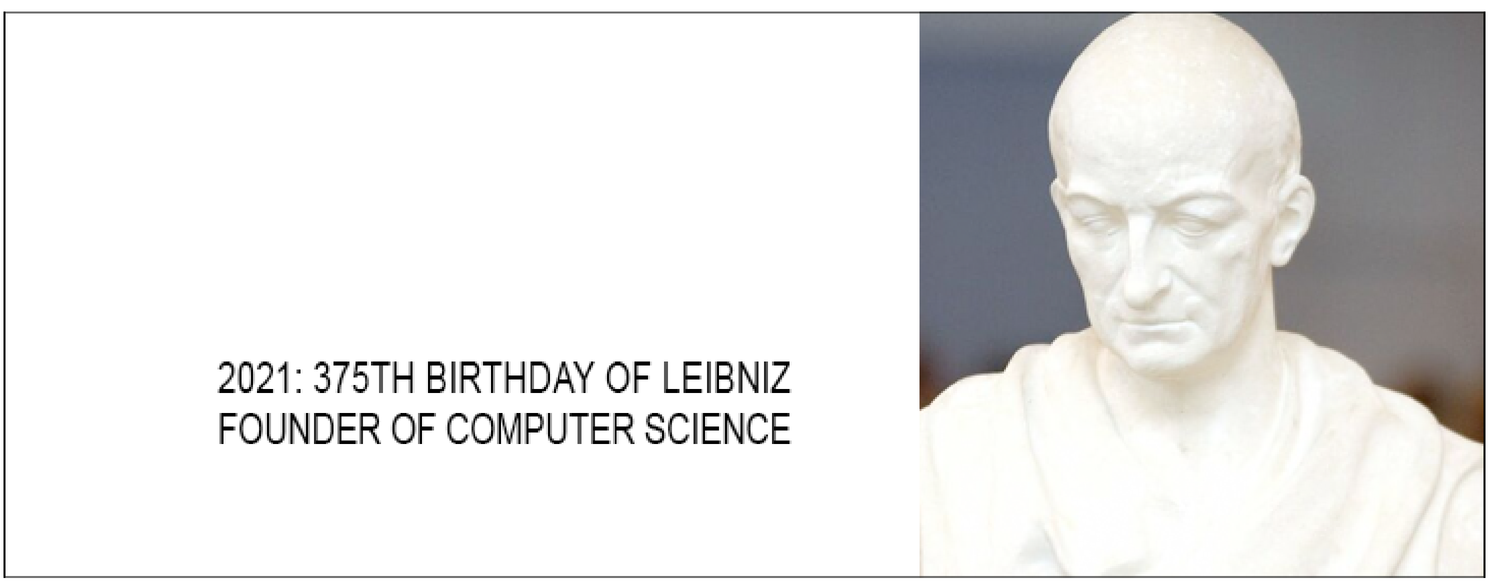
The recent breakthroughs of deep learning algorithms from the past millennium (see previous sections) would have been impossible without continually improving and accelerating computer hardware. Any history of AI and deep learning would be incomplete without mentioning this evolution, which has been running for at least two millennia.

The first known gear-based computational device was the Antikythera mechanism (a kind of astronomical clock) in Ancient Greece over 2000 years ago.

Perhaps the world's first *practical programmable machine* was an automatic theatre made in the 1st century[SHA7a][RAU1] by Heron of Alexandria (who apparently also had the first known working steam engine—the *Aeolipile*).



The 9th century music automaton by the Banu Musa brothers in Baghdad was perhaps the first machine with a *stored* program.[BAN][KOE1] It used pins on a revolving cylinder to store programs controlling a steam-driven flute—compare Al-Jazari's programmable drum machine of 1206. [SHA7b]

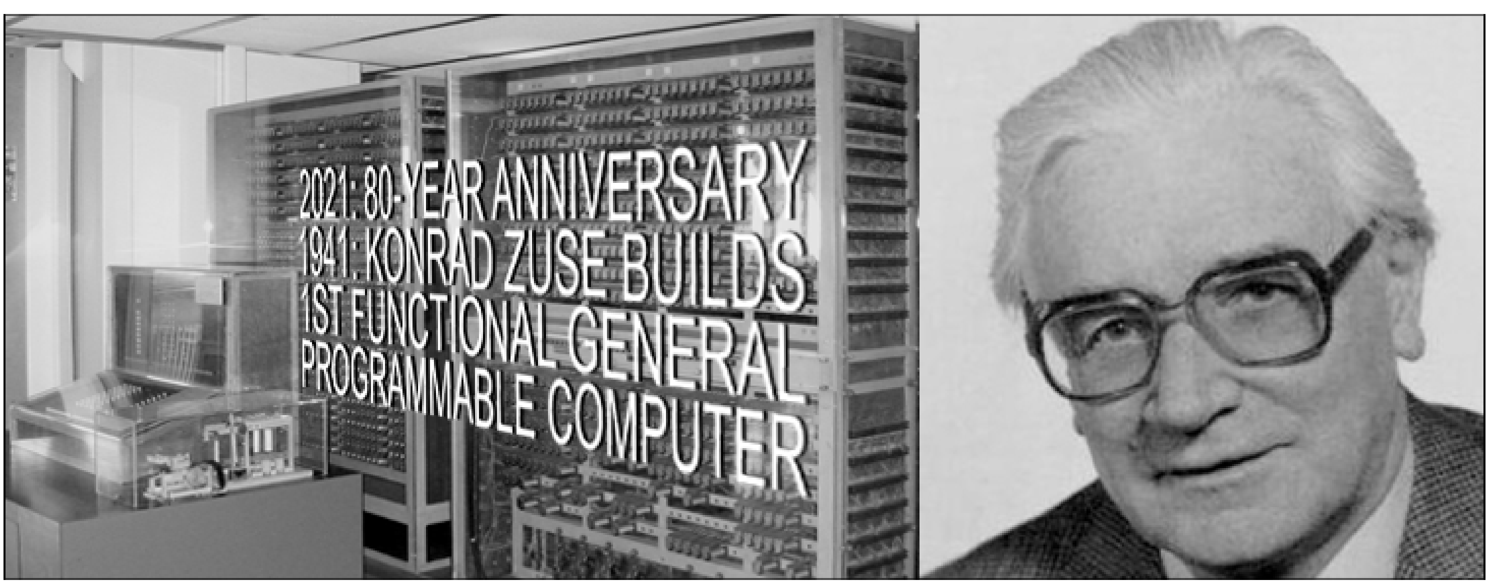


The 1600s brought more flexible machines that computed answers in response to *input data*. The first data-processing gear-based special purpose calculator for simple arithmetics was built in 1623 by Wilhelm Schickard, one of the candidates for the title of "father of automatic computing," followed by the superior Pascaline of Blaise Pascal (1642). In 1673, the already mentioned Gottfried Wilhelm Leibniz (called "the smartest man who ever lived"[SMO13]) designed the first machine (the step reckoner) that could perform all four arithmetic operations, and the first with a memory.[BL16] He also described the principles of binary computers governed by punch cards (1679),[L79][L03][LA14][HO66] and published the chain rule[LEI07-10] (see above), essential ingredient of deep learning and modern AI.

The first *commercial* program-controlled machines (punch card-based looms) were built in France circa 1800 by Joseph-Marie Jacquard and others—perhaps the first "modern" programmers who wrote the world's first *industrial* software. They inspired Ada Lovelace and her mentor Charles Babbage (UK, circa 1840). He planned but was unable to build a programmable, general purpose computer (only his *non-universal special purpose calculator* led to a working 20th century replica).

In 1914, the Spaniard Leonardo Torres y Quevedo (mentioned in the introduction) became the 20th century's first AI pioneer when he built the first working chess end game player (back then chess was considered as an activity restricted to the realms of intelligent creatures). The machine was still considered impressive decades later when another AI pioneer—Norbert Wiener[WI48]—played against it at the 1951 Paris AI conference.[AI51][BRO21][BRU4]

Between 1935 and 1941, Konrad Zuse created the world's first working programmable general-purpose computer: the Z3. The corresponding patent of 1936[ZU36-38][RO98][ZUS21] described the digital circuits required by programmable physical hardware, predating Claude Shannon's 1937 thesis on digital circuit design.[SHA37] Unlike Babbage, Zuse used Leibniz' principles of *binary computation* (1679)[L79][LA14][HO66][L03] instead of traditional *decimal computation*. This greatly simplified the hardware.[LEI21,a,b] Ignoring the inevitable storage limitations of any physical computer, the *physical hardware* of Z3 was indeed *universal* in the modern sense of the *purely theoretical but impractical* constructs of Gödel[GOD][GOD34,21,21a] (1931-34), Church[CHU] (1935), Turing[TUR] (1936), and Post[POS] (1936). Simple arithmetic tricks can compensate for Z3's lack of an explicit conditional jump instruction.[RO98] Today, most computers are *binary* like Z3.



Z3 used *electromagnetic* relays with visibly moving switches. The first *electronic* special purpose calculator (whose moving parts were electrons too small to see) was the binary ABC (US, 1942) by John Atanasoff (the "father of tube-based computing"[NASC6a]). Unlike the gear-based machines of the 1600s, ABC used vaccum tubes—today's machines use the transistor principle patented by Julius Edgar Lilienfeld in 1925.[LIL1-2] But unlike Zuse's Z3, ABC was not freely programmable. Neither was the *electronic* Colossus machine by Tommy Flowers (UK, 1943-45) used to break the Nazi code.[NASC6]

The first general working programmable machine built by someone other than Zuse (1941)[RO98] was Howard Aiken's decimal MARK I (US, 1944). The much faster decimal ENIAC by Eckert and Mauchly (1945/46) was programmed by rewiring it. Both data *and* programs were stored in electronic memory by the "Manchester baby" (Williams, Kilburn & Tootill, UK, 1948) and the 1948 upgrade of ENIAC, which was reprogrammed by entering numerical instruction codes into read-only memory.[HAI14b]

Since then, computers have become much faster through integrated circuits (ICs). In 1949, Werner Jacobi at Siemens filed a patent for an IC semiconductor with several transistors on a common substrate (granted in 1952).[IC49-14] In 1958, Jack Kilby demonstrated an IC with external wires. In 1959, Robert Noyce presented a monolithic IC.[IC14] Since the 1970s, graphics processing units (GPUs) have been used to speed up computations through parallel processing. ICs/GPUs of today (2022) contain many billions of transistors (almost all of them of Lilienfeld's 1925 FET type[LIL1-2]).

In 1941, Zuse's Z3 could perform roughly one elementary operation (e.g., an addition) per second. Since then, every 5 years, compute got 10 times cheaper (note that his law is much older than Moore's Law which states that the number of transistors[LIL1-2] per chip doubles every 18 months). As of 2021, 80 years after Z3, modern computers can execute about 10 million billion instructions per second for the same (inflation-adjusted) price. The naive extrapolation of this exponential trend predicts that the 21st century will see cheap computers with a thousand times the raw computational power of all human brains combined.[RAW]

Where are the physical limits? According to Bremermann (1982),[BRE] a computer of 1 kg of mass and 1 liter of volume can execute at most 10^{51} operations per second on at most 10^{32} bits. The trend above will hit the Bremermann limit roughly 25 decades after Z3, circa 2200. However, since there are only 2 x 10^{30} kg of mass in the solar system, the trend is bound to break within a few centuries, since the speed of light will greatly limit the acquisition of additional mass, e.g., in form of other solar systems, through a function ploynomial in time, as previously noted back in 2004.[OOPS2][ZUS21]

Physics seems to dictate that future efficient computational hardware will have to be brain-like, with many compactly placed processors in 3-dimensional space, sparsely connected by many short and few long wires, to minimize total connection cost (even if the "wires" are actually light beams).[DL2] The basic architecture is essentially the one of a deep, sparsely connected, 3-dimensional RNN, and Deep Learning methods for such RNNs are expected to become even much more important than they are today.[DL2]

Don't Neglect the Theory of AI Since 1931

The core of modern AI and deep learning is mostly based on simple math of recent centuries: calculus/linear algebra/statistics. Nevertheless, to efficiently implement this core on the modern hardware mentioned in the previous section, and to roll it out for billions of people, lots of software engineering was necessary, based on lots of smart algorithms invented in the past century. There is no room here to mention them all. However, at least I'll list some of the most important highlights of the theory of AI and computer science in general.

In the early 1930s, Gödel founded modern theoretical computer science.[GOD][GOD34][LEI21,21a] He introduced a *universal coding language* (1931-34).[GOD][GOD34-21a] It was based on the integers, and allows for formalizing the operations of any digital computer in axiomatic form. Gödel used it to represent both data (such as axioms and theorems) and programs[VAR13] (such as proof-generating sequences of operations on the data). He famously constructed formal statements that talk about the computation of other formal statements—especially self-referential statements which imply that they are not decidable, given a computational theorem prover that systematically enumerates all possible theorems from an enumerable set of axioms. Thus he identified fundamental limits of algorithmic theorem proving, computing, and any type of computation-based AI.[GOD][BIB3][MIR](Sec. 18)[GOD21,21a]

Like most great scientists, Gödel built on earlier work. He combined Georg Cantor's diagonalization trick[CAN] (which showed in 1891 that there are different types of infinities) with the foundational work by Gottlob Frege[FRE] (who introduced the first formal language in 1879), Thoralf Skolem[SKO23] (who introduced primitive recursive functions in 1923) and Jacques Herbrand[GOD86] (who identified limitations of Skolem's approach). These authors in turn built on the formal *Algebra of Thought* (1686) by Gottfried Wilhelm Leibniz[L86][WI48] (see above), which is deductively equivalent[LE18] to the later *Boolean Algebra* of 1847.[BOO]

In 1935, Alonzo Church derived a corollary / extension of Gödel's result by demonstrating that Hilbert & Ackermann's *Entscheidungsproblem* (decision problem) does not have a general solution.[CHU] To do this, he used his alternative universal coding language called *Untyped Lambda Calculus*, which forms the basis of the highly influential programming language LISP.

In 1936, Alan M. Turing introduced yet another universal model: the *Turing Machine*.[TUR] He rederived the above-mentioned result.[CHU][TUR][HIN][GOD21,21a][TUR21][LEI21,21a] In the same year of 1936, Emil Post published yet another independent universal model of computing.[POS] Today we know many such models.

Konrad Zuse not only created the world's first working programmable general-purpose computer,[ZU36-38][RO98][ZUS21] he also designed *Plankalkül*, the first high-level programming language.[BAU][KNU] He applied it to chess in 1945[KNU] and to theorem proving in 1948.[ZU48] Compare Newell & Simon's later work on theorem proving (1956).[NS56] Much of early AI in the 1940s-70s was actually about theorem proving and deduction in Gödel style[GOD][GOD34,21,21a] through expert systems and logic programming.

In 1964, Ray Solomonoff combined Bayesian (actually Laplacian[STI83-85]) probabilistic reasoning and theoretical computer science[GOD][CHU][TUR][POS] to derive a mathematically optimal (but computationally infeasible) way of learning to predict future data from past observations.[AIT1][AIT10] With Andrej Kolmogorov, he founded the theory of Kolmogorov complexity or algorithmic information theory (AIT),[AIT1-22] going beyond traditional information theory[SHA48][KUL] by formalizing the concept of Occam's razor, which favors the simplest explanation of given data, through the concept of the shortest program computing the data. There are many computable, time-bounded versions of this concept,[AIT7][AIT5][AIT12-13][AIT16-17] as well as applications to NNs.[KO2][CO1-3]

In the early 2000s, Marcus Hutter (while working under my Swiss National Science Foundation grant[UNI]) augmented Solomonoff's universal predictor[AIT1][AIT10] by an optimal action selector (a universal AI) for reinforcement learning agents living in initially unknown (but at least computable) environments.[AIT20,22] He also derived the asymptotically fastest algorithm for all well-defined computational problems,[AIT21] solving any problem as quickly as the unknown fastest solver of such problems, save for an additive constant that does not depend on the problem size.

The even more general optimality of the self-referential 2003 Gödel Machine[GM3-9] is not limited to *asymptotic* optimality.

Nevertheless, such mathematically optimal AIs are not yet practically feasible for various reasons. Instead, practical modern AI is based on suboptimal, limited, yet not extremely well-understood techniques such as NNs and deep learning, the focus of the present article. But who knows what kind of AI history will prevail 20 years form now?

The Broader Historic Context from Big Bang to Far Future

Credit assignment is about finding patterns in historic data and figuring out how certain events were enabled by previous events. Historians do it. Physicists do it. AIs do it, too. Let's take a step back and look at AI in the broadest historical context: all time since the Big Bang. In 2014, I found a beautiful pattern of exponential acceleration in it,[OMG] which I have presented in many talks since then, and which also made it into Sibylle Berg's award-winning book *"GRM:*

Brainfuck."[OMG2] Previously published patterns of this kind span much shorter time intervals: just a few decades or centuries or at most millennia.[OMG1]

It turns out that from a human perspective, the most important events since the beginning of the universe are neatly aligned on a timeline of exponential speed up (error bars mostly below 10 percent). In fact, history seems to converge in an Omega point in the year 2040 or so. I like to call it Omega, because a century ago, Teilhard de Chardin called Omega the point where humanity will reach its next level.[OMG0] Also, Omega sounds much better than "Singularity"[SING1-2]—it sounds a bit like "Oh my God."[OMG]

Let's start with the Big Bang 13.8 billion years ago. We divide this time by 4 to obtain about 3.5 billion years. Omega is 2040 or so. At Omega minus 3.5 billion years, something very important happened: life emerged on this planet.

And we take again a quarter of this time. We come out 900 million years ago when something very important happened: animal-like, mobile life emerged.

And we divide again by 4. We come out 220 million years ago when mammals were invented, our ancestors.

And we divide again by 4. 55 million years ago the first primates emerged, our ancestors.

And we divide again by 4. 13 million years ago the first hominids emerged, our ancestors. I don't know why all these divisions by 4 keep hitting these defining moments in history. But they do. I also tried thirds, and fifths, and harmonic proportions, but only quarters seem to work.

And we divide again by 4. 3.5 million years ago something very important happened: the dawn of technology, as *Nature* called it: the first stone tools.

And we divide by 4. 800 thousand years ago the next great tech breakthrough happened: controlled fire.

And we divide by 4. 200 thousand years ago, anatomically modern man became prominent, our ancestor.

And we divide by 4. 50 thousand years ago, behaviorally modern man emerged, our ancestor, and started colonizing the world.

And we divide again by 4. We come out 13 thousand years ago when something very important happened: domestication of animals, agriculture, first settlements—the begin of civilisation. Now we see that all of civilization is just a flash in world history, just one millionth of the time since the Big Bang. Agriculture and spacecraft were invented almost at the same time.

And we divide by 4. 3,300 years ago saw the onset of the 1st population explosion in the Iron Age.

And we divide by 4. Remember that the convergence point Omega is the year 2040 or so. Omega minus 800 years—that was in the 13th century, when iron and fire came together in form of guns and cannons and rockets in China. This has defined the world since then and the West remains quite behind of the license fees it owes China.

And we divide again by 4. Omega minus 200 years—we hit the mid 19th century, when iron and fire came together in ever more sophisticated form to power the industrial revolution through improved steam engines, based on the work of Beaumont, Papin, Newcomen, Watt, and others (1600s-1700s, going beyond the first simple steam engines by Heron of Alexandria[RAU1] in the 1st century). The telephone (e.g., Meucci 1857, Reis 1860, Bell 1876)[NASC3] started to revolutionize communication. The germ theory of disease (Pasteur & Koch, late 1800s) revolutionized healthcare and made people live longer on average. And circa 1850, the fertilizer-based agricultural revolution (Sprengel & von Liebig, early 1800s) helped to trigger the 2nd population explosion, which peaked in the 20th century, when the world population quadrupled, letting the 20th century stand out among all centuries in the history of mankind, driven by the Haber-Bosch process for creating *artificial* fertilizer, without which the world could feed at most 4 billion people.[HAB1-2]

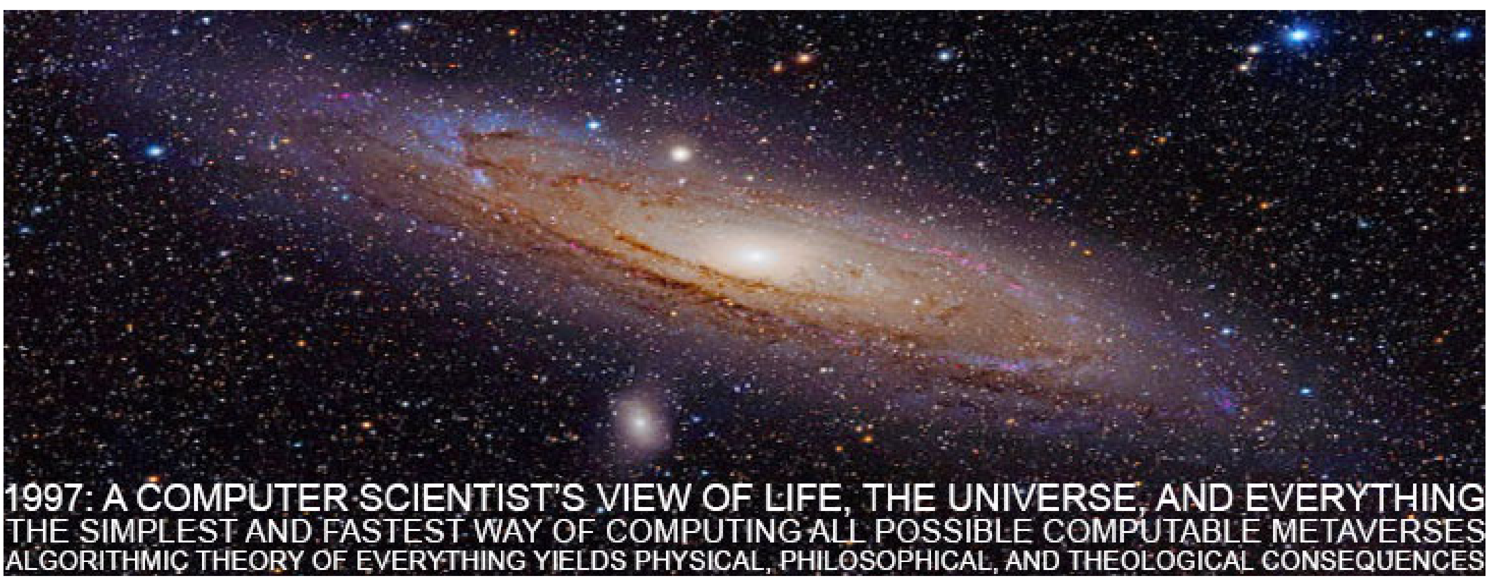
And we divide again by 4. Omega minus 50 years—that's more or less the year 1990, the end of the 3 great wars of the 20th century: WW1, WW2, and the Cold War. The 7 most valuable public companies were all Japanese (today most of them are US-based); however, both China and the US West Coast started to rise rapidly, setting the stage for the 21st century. A digital nervous system started spanning the globe through cell phones and the wireless revolution (based on radio waves discovered in the 1800s) as well as cheap personal computers for all. The WWW was created at the European particle collider in Switzerland by Tim Berners-Lee. And Modern AI started also around this time: the first truly self-driving cars were built in the 1980s in Munich by the team of Ernst Dickmanns (by 1994, their robot cars were driving in highway traffic, up to 180 km/h).[AUT] Back then, I worked on my 1987 diploma thesis,[META1] which introduced algorithms not just for learning but also for meta-learning or learning to learn,[META] to learn better learning algorithms through experience (now a very popular topic[DEC]). And then came our Miraculous Year 1990-91[MIR] at TU Munich, the root of today's most cited NNs[MOST] and of modern deep learning through self-supervised/unsupervised learning (see above),[UN][UN0-3] the LSTM/Highway Net/ResNet principle (now in your pocket on your smartphone—see above),[DL4][DEC][MOST] artificial curiosity and generative adversarial NNs for agents that invent their own problems (see above),[AC90-AC20][PP-PP2][SA17] Transformers with linearized self-attention (see above),[FWP0-6][TR5-6] distilling teacher NNs into student NNs (see above),[UN][UN0-3] learning action plans at multiple levels of abstraction and multiple time scales (see above),[HRL0-2][LEC] and other exciting stuff. Much of this has become very popular, and improved the lives of billions of people.[DL4][DEC][MOST]

And we divide again by 4. Omega minus 13 years—that's a point in the near future, more or less the year 2030, when many predict that cheap AIs will have a human brain power. Then the final 13 years or so until Omega, when incredible things will happen (take all of this with a grain of salt, though[OMG1]).

But of course, time won't stop with Omega. Maybe it's just human-dominated history that will end. After Omega, many curious meta-learning AIs that invent their own goals (which have existed in my lab for decades[AC][AC90,AC90b]) will quickly improve themselves, restricted only by the fundamental limits of computability and physics.

What will supersmart AIs do? Space is hostile to humans but friendly to appropriately designed robots, and offers many more resources than our thin film of biosphere, which receives less than a billionth of the sun's energy. While some curious AIs will remain fascinated with life, at least as long as they don't fully understand it,[ACM16][FA15][SP16][SA17] most will be more interested in the incredible new opportunities for robots and software life out there in space. Through innumerable self-replicating robot factories in the asteroid belt and beyond they will transform the solar system and then within a few hundred thousand years the entire galaxy and within tens of billions of years the rest of the reachable universe. Despite the light-speed limit, the expanding AI sphere will have plenty of time to colonize and shape the entire visible cosmos.

Let me stretch your mind a bit. The universe is still young, only 13.8 billion years old. Remember when we kept dividing by 4? Now let's multiply by 4! Let's look ahead to a time when the cosmos will be 4 times older than it is now: about 55 billion years old. By then, the visible cosmos will be permeated by intelligence. Because after Omega, most AIs will have to go where most of the physical resources are, to make more and bigger AIs. Those who don't won't have an impact.[ACM16][FA15][SP16]



Acknowledgments

Some of the material above was taken from previous AI Blog posts.[MIR] [DEC] [GOD21] [ZUS21] [LEI21] [AUT] [HAB2] [ARC06] [AC] [ATT] [DAN] [DAN1] [DL4] [GPUCNN5,8] [DLC] [FDL] [FWP] [LEC] [META] [MLP2] [MOST] [PLAN]

[UN] [LSTMPG] [BP4] [DL6a] [HIN] [T22] Thanks to many expert reviewers (including several famous neural net pioneers) for useful comments. Since science is about self-correction, let me know under *juergen@idsia.ch* if you can spot any remaining error. Many additional relevant publications can be found in my publication page and my arXiv page.

555+ References (and many more in the survey[DL1])

[25y97] In 2022, we are celebrating the following works from a quarter-century ago. 1. Journal paper on Long Short-Term Memory, the most cited neural network (NN) of the 20th century (and basis of the most cited NN of the 21st). 2. First paper on physical, philosophical and theological consequences of the simplest and fastest way of computing all possible metaverses (= computable universes). 3. Implementing artificial curiosity and creativity through generative adversarial agents that learn to design *abstract, interesting* computational experiments. 4. Journal paper on meta-reinforcement learning. 5. Journal paper on hierarchical Q-learning. 6. First paper on reinforcement learning to play soccer: start of a series. 7. Journal papers on flat minima & low-complexity NNs that generalize well. 8. Journal paper on Low-Complexity Art, the Minimal Art of the Information Age. 9. Journal paper on probabilistic incremental program evolution.

[AC] J. Schmidhuber (AI Blog, 2021). 3 decades of artificial curiosity & creativity. *Schmidhuber's artificial scientists not only answer given questions but also invent new questions. They achieve curiosity through: (1990) the principle of generative adversarial networks, (1991) neural nets that maximise learning progress, (1995) neural nets that maximise information gain (optimally since 2011), (1997) adversarial design of surprising computational experiments, (2006) maximizing compression progress like scientists/artists/comedians do, (2011) PowerPlay... Since 2012: applications to real robots.*

[AC90] J. Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. Technical Report FKI-126-90, TUM, Feb 1990, revised Nov 1990. PDF. *The first paper on online planning with reinforcement learning recurrent neural networks (NNs) (more) and on generative adversarial networks where a generator NN is fighting a predictor NN in a minimax game (more).*

[AC90b] 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. Based on [AC90]. PDF. More.

[AC91] J. Schmidhuber. Adaptive confidence and adaptive curiosity. Technical Report FKI-149-91, Inst. f. Informatik, Tech. Univ. Munich, April 1991. PDF.

[AC91b] J. Schmidhuber. Curious model-building control systems. *Proc. International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458-1463. IEEE, 1991. PDF.

[AC97] J. Schmidhuber. What's interesting? Technical Report IDSIA-35-97, IDSIA, July 1997. *Focus on automatic creation of predictable internal abstractions of complex spatio-temporal events: two competing, intrinsically motivated agents agree on essentially arbitrary algorithmic experiments and bet on their possibly surprising (not yet predictable) outcomes in zero-sum games, each agent potentially profiting from outwitting / surprising the other by inventing experimental protocols where both modules disagree on the predicted outcome. The focus is on exploring the space of general algorithms (as opposed to traditional simple mappings from inputs to outputs); the general system focuses on the interesting things by losing interest in both predictable and unpredictable aspects of the world. Unlike Schmidhuber et al.'s previous systems with intrinsic motivation,[AC90-AC95] the system also takes into account the computational cost of learning new skills, learning when to learn and what to learn. See later publications.[AC99][AC02]*

[AC98] M. Wiering and J. Schmidhuber. Efficient model-based exploration. In R. Pfeiffer, B. Blumberg, J. Meyer, S. W. Wilson, eds., *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, p. 223-228, MIT Press, 1998.

[AC98b] M. Wiering and J. Schmidhuber. Learning exploration policies with models. In *Proc. CONALD*, 1998.

[AC99] J. Schmidhuber. Artificial Curiosity Based on Discovering Novel Algorithmic Predictability Through Coevolution. In P. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, Z. Zalzala, eds., Congress on Evolutionary Computation, p. 1612-1618, IEEE Press, Piscataway, NJ, 1999.

[AC02] J. Schmidhuber. Exploring the Predictable. In Ghosh, S. Tsutsui, eds., Advances in Evolutionary Computing, p. 579-612, Springer, 2002. PDF.

[AC06] J. Schmidhuber. Developmental Robotics, Optimal Artificial Curiosity, Creativity, Music, and the Fine Arts. *Connection Science,* 18(2): 173-187, 2006. PDF.

[AC09] J. Schmidhuber. Art & science as by-products of the search for novel patterns, or data compressible in unknown yet learnable ways. In M. Botta (ed.), Et al. Edizioni, 2009, pp. 98-112. PDF. (More on artificial scientists and artists.)

[AC10] J. Schmidhuber. Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230-247, 2010. IEEE link. PDF. *With a brief summary of the generative adversarial neural networks of 1990[AC90,90b][AC20] where a generator NN is fighting a predictor NN in a minimax game (more).*

[AC20] J. Schmidhuber. Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991). Neural Networks, Volume 127, p 58-66, 2020. Preprint arXiv/1906.04493.

[ACM16] ACM interview by S. Ibaraki (2016). Chat with J. Schmidhuber: Artificial Intelligence & Deep Learning—Now & Future. Link.

[AIB] J. Schmidhuber. AI Blog. Includes variants of chapters of the AI Book.

[AI51] Les Machines a Calculer et la Pensee Humaine: Paris, 8.-13. Januar 1951, Colloques internationaux du Centre National de la Recherche Scientifique; no. 37, Paris 1953. *H. Bruderer[BRU4] calls that the first conference on AI.*

[AIT1] R. J. Solomonoff. A formal theory of inductive inference. Part I. Information and Control, 7:1-22, 1964.

[AIT2] A. N. Kolmogorov. Three approaches to the quantitative definition of information. Problems of Information Transmission, 1:1-11, 1965.

[AIT3] G.J. Chaitin. On the length of programs for computing finite binary sequences: statistical considerations. Journal of the ACM, 16:145-159, 1969 (submitted 1965).

[AIT4] P. Martin-Löf. The definition of random sequences. Information and Control, 9:602-619, 1966.

[AIT5] C. S. Wallace and D. M. Boulton. An information theoretic measure for classification. Computer Journal, 11(2):185-194, 1968.

[AIT6] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the algorithmic concepts of information and randomness. Russian Math. Surveys, 25(6):83-124, 1970.

[AIT7] L. A. Levin. Universal sequential search problems. Problems of Information Transmission, 9(3):265-266, 1973.

[AIT8] L. A. Levin. Laws of information (nongrowth) and aspects of the foundation of probability theory. Problems of Information Transmission, 10(3):206-210, 1974.

[AIT9] C. P. Schnorr. Process complexity and effective random tests. Journal of Computer Systems Science, 7:376-388, 1973.

[AIT10] R.J. Solomonoff. Complexity-based induction systems. IEEE Transactions on Information Theory, IT-24(5):422-432, 1978.

[AIT11] P. Gacs. On the relation between descriptional complexity and algorithmic probability. Theoretical Computer Science, 22:71-93, 1983.

[AIT12] J. Hartmanis. Generalized Kolmogorov complexity and the structure of feasible computations. Proc. 24th IEEE Symposium on Foundations of Computer Science, pages 439-445, 1983.

[AIT13] J. Rissanen. Stochastic complexity and modeling. The Annals of Statistics, 14(3):1080-1100, 1986.

[AIT14] Y. M. Barzdin. Algorithmic information theory. In D. Reidel, editor, Encyclopaedia of Mathematics, vol. 1, pages 140-142. Kluwer Academic Publishers, 1988.

[AIT15] O. Watanabe. Kolmogorov complexity and computational complexity. EATCS Monographs on Theoretical Computer Science, Springer, 1992.

[AIT16] M. Li and P. M. B. Vitanyi. An Introduction to Kolmogorov Complexity and its Applications (2nd edition). Springer, 1997.

[AIT17] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. Neural Networks, 10(5):857-873, 1997.

[AIT18] M. S. Burgin and Y. M. Borodyanskii. Infinite processes and super-recursive algorithms. Notices of the Academy of Sciences of the USSR (translated from Russian), 321(5):800-803, 1991.

[AIT19] C. S. Calude. Chaitin Omega numbers, Solovay machines and Gödel incompleteness. Theoretical Computer Science, 2000.

[AIT20] M. Hutter. A theory of universal artificial intelligence based on algorithmic complexity. Technical Report IDSIA-14-00 (cs.AI/0004001), IDSIA, Manno (Lugano), CH, 2000.

[AIT21] M. Hutter. The fastest and shortest algorithm for all well-defined problems. International Journal of Foundations of Computer Science, 13(3):431-443, 2002. (Based on work done under J. Schmidhuber's SNF grant 20-61847: unification of universal induction and sequential decision theory, 2000).

[AIT22] M. Hutter. Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer, Berlin, 2004. (Based on work done under J. Schmidhuber's SNF grant 20-61847: unification of universal induction and sequential decision theory, 2000).

[AIT23] 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.

[AIT24] 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.

[AM16] Blog of Werner Vogels, CTO of Amazon (Nov 2016): Amazon's Alexa *"takes advantage of bidirectional long short-term memory (LSTM) networks using a massive amount of data to train models that convert letters to sounds and predict the intonation contour. This technology enables high naturalness, consistent intonation, and accurate processing of texts."*

[AMH0] S. I. Amari (1972). Characteristics of random nets of analog neuron-like elements. IEEE Trans. Syst. Man Cybernetics, 2, 643-657. *First published 1969 in Japanese, long before Wilson & Cowan's very similar work (1972-73).*

[AMH1] S. I. Amari (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. IEEE Transactions, C 21, 1197-1206, 1972. PDF. *First publication of what was later sometimes called the Hopfield network*[AMH2] *or Amari-Hopfield Network,*[AMH3] *based on the (uncited) Lenz-Ising recurrent architecture.*[L20][I25][T22]

[AMH1b] W. A. Little. The existence of persistent states in the brain. Mathematical Biosciences, 19.1-2, p. 101-120, 1974. *Mentions the recurrent Ising model[L20][I25]on which the (uncited) Amari network[AMH1,2] is based.*

[AMH2] J. J. Hopfield (1982). Neural networks and physical systems with emergent collective computational abilities. Proc. of the National Academy of Sciences, vol. 79, pages 2554-2558, 1982. *The Hopfield network or Amari-Hopfield Network was first published in 1972 by Amari.[AMH1] [AMH2] did not cite [AMH1].*

[AMH3] A. P. Millan, J. J. Torres, J. Marro. How Memory Conforms to Brain Development. Front. Comput. Neuroscience, 2019

[ARC06] J. Schmidhuber (2006). Archimedes—Greatest Scientist Ever?

[ATT] J. Schmidhuber (AI Blog, 2020). 30-year anniversary of end-to-end differentiable sequential neural attention. Plus goal-conditional reinforcement learning. *Schmidhuber had both hard attention for foveas (1990) and soft attention in form of Transformers with linearized self-attention (1991-93).[FWP] Today, both types are very popular.*

[ATT0] J. Schmidhuber and R. Huber. Learning to generate focus trajectories for attentive vision. Technical Report FKI-128-90, Institut für Informatik, Technische Universität München, 1990. PDF.

[ATT1] J. Schmidhuber and R. Huber. Learning to generate artificial fovea trajectories for target detection. International Journal of Neural Systems, 2(1 & 2):135-141, 1991. Based on TR FKI-128-90, TUM, 1990. PDF. More.

[ATT2] J. Schmidhuber. Learning algorithms for networks with internal and external feedback. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Proc. of the 1990 Connectionist Models Summer School*, pages 52-61. San Mateo, CA: Morgan Kaufmann, 1990. PS. (PDF.)

[ATT3] H. Larochelle, G. E. Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. NIPS 2010. *This work is very similar to [ATT0-2] which the authors did not cite. In fact, Hinton was the reviewer of a 1990 paper[ATT2] which summarised in its Section 5 Schmidhuber's early work on attention: the first implemented neural system for combining glimpses that jointly trains a recognition & prediction component with an attentional component (the fixation controller). Two decades later, Hinton wrote about his own work:[ATT3] "To our knowledge, this is the first implemented system for combining glimpses that jointly trains a recognition component ... with an attentional component (the fixation controller)." See [MIR] (Sec. 9)[R4].*

[ATT14] D. Bahdanau, K. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. 2014-16. Preprint arXiv/1409.0473, 2014-16. *This work on soft "attention" did not cite Schmidhuber's much earlier original work of 1991-1993 on soft attention and Transformers with linearized self-attention.[FWP,FWP0-2,6][ATT]*

[AUT] J. Schmidhuber (AI Blog, 2005). Highlights of robot car history. *Around 1986, Ernst Dickmanns and his group at Univ. Bundeswehr Munich built the world's first real autonomous* *robot cars, using saccadic vision, probabilistic approaches such as Kalman filters, and parallel computers. By 1994, they were in highway traffic, at up to 180 km/h, automatically passing other cars.*

[AV1] A. Vance. Google Amazon and Facebook Owe Jürgen Schmidhuber a Fortune—This Man Is the Godfather the AI Community Wants to Forget. Business Week, Bloomberg, May 15, 2018.

[BA93] P. Baldi and Y. Chauvin. Neural Networks for Fingerprint Recognition, Neural Computation, Vol. 5, 3, 402-418, (1993). *First application of CNNs with backpropagation to biomedical/biometric images.*

[BA96] P. Baldi and Y. Chauvin. Hybrid Modeling, HMM/NN Architectures, and Protein Applications, Neural Computation, Vol. 8, 7, 1541-1565, (1996). *One of the first papers on graph neural networks.*

[BA99] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the Past and the Future in Protein Secondary Structure Prediction, Bioinformatics, Vol. 15, 11, 937-946, (1999).

[BA03] P. Baldi and G. Pollastri. The Principled Design of Large-Scale Recursive Neural Network Architectures-DAG-RNNs and the Protein Structure Prediction Problem. Journal of Machine Learning Research, 4, 575-602, (2003).

[BAN] Banu Musa brothers (9th century). The book of ingenious devices (Kitab al-hiyal). Translated by D. R. Hill (1979), Springer, p. 44, ISBN 90-277-0833-9. *As far as we know, the Banu Musa music automaton was the world's first machine with a stored program.*

[BAU] F. L. Bauer, H. Woessner (1972). The "Plankalkül" of Konrad Zuse: A Forerunner of Today's Programming Languages.

[BAY1] S. M. Stigler. Who Discovered Bayes' Theorem? The American Statistician. 37(4):290-296, 1983. *Bayes' theorem is actually Laplace's theorem or possibly Saunderson's theorem.*

[BAY2] T. Bayes. An essay toward solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society of London, 53:370-418. Communicated by R. Price, in a letter to J. Canton, 1763.

[BAY3] D. J. C. MacKay. A practical Bayesian framework for backprop networks. Neural Computation, 4:448-472, 1992.

[BAY4] J. Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Synthese-Dordrecht 104(1):161, 1995.

[BAY5] N. M. Oliver, B. Rosario, A. P. Pentland A Bayesian computer vision system for modeling human interactions. IEEE transactions on pattern analysis and machine intelligence 22.8:831-843, 2000.

[BAY6] D. M. Blei, A. Y. Ng, M. I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research 3:993-1022, 2003.

[BAY7] J. F. G. De Freitas. Bayesian methods for neural networks. PhD thesis, University of Cambridge, 2003.

[BAY8] J. M. Bernardo, A.F.M. Smith. Bayesian theory. Vol. 405. John Wiley & Sons, 2009.

[BB2] J. Schmidhuber. A local learning algorithm for dynamic feedforward and recurrent networks. Connection Science, 1(4):403-412, 1989. (The Neural Bucket Brigade—figures omitted!). PDF. HTML. Compare TR FKI-124-90, TUM, 1990. PDF. *Proposal of a biologically more plausible deep learning algorithm that—unlike backpropagation—is local in space and time. Based on a "neural economy" for reinforcement learning.*

[BEL53] R. Bellman. An introduction to the theory of dynamic programming. RAND Corp. Report, 1953

[BIB3] W. Bibel (2003). Mosaiksteine einer Wissenschaft vom Geiste. Invited talk at the conference on AI and Gödel, Arnoldsheim, 4-6 April 2003. Manuscript, 2003.

[BL16] L. Bloch (2016). Informatics in the light of some Leibniz's works. Communication to XB2 Berlin Xenobiology Conference.

[BM] D. Ackley, G. Hinton, T. Sejnowski (1985). A Learning Algorithm for Boltzmann Machines. Cognitive Science, 9(1):147-169. *This paper neither cited relevant prior work by Sherrington & Kirkpatrick*[SK75] *& Glauber*[G63] *nor the first working algorithms for deep learning of internal representations (Ivakhnenko & Lapa, 1965)*[DEEP1-2][HIN] *nor Amari's work (1967-68)*[GD1-2] *on learning internal representations in deep nets through stochastic gradient descent. Even later surveys by the authors*[S20][DLC] *failed to cite the prior art.*[T22]

[BER96] D. P. Bertsekas, J. N. Tsitsiklis. Neuro-dynamic Programming. Athena Scientific, Belmont, MA, 1996.

[BOO] George Boole (1847). The Mathematical Analysis of Logic, Being an Essay towards a Calculus of Deductive Reasoning. London, England: Macmillan, Barclay, & Macmillan, 1847. Leibniz' formal *Algebra of Thought* (1686)[L86][WI48] was deductively equivalent[LE18] to the much later *Boolean Algebra*.

[BOU] H Bourlard, N Morgan (1993). Connectionist speech recognition. Kluwer, 1993.

[BPA] H. J. Kelley. Gradient Theory of Optimal Flight Paths. ARS Journal, Vol. 30, No. 10, pp. 947-954, 1960. *Precursor of modern backpropagation.*[BP1-5]

[BPB] A. E. Bryson. A gradient method for optimizing multi-stage allocation processes. Proc. Harvard Univ. Symposium on digital computers and their applications, 1961.

[BPC] S. E. Dreyfus. The numerical solution of variational problems. Journal of Mathematical Analysis and Applications, 5(1): 30-45, 1962.

[BP1] S. Linnainmaa. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki, 1970. *See chapters 6-7 and FORTRAN code on pages 58-60.* PDF. See also BIT 16, 146-160,

1976. Link. *The first publication on "modern" backpropagation, also known as the reverse mode of automatic differentiation.*

[BP2] P. J. Werbos. Applications of advances in nonlinear sensitivity analysis. In R. Drenick, F. Kozin, (eds): System Modeling and Optimization: Proc. IFIP, Springer, 1982. PDF. *First application of backpropagation[BP1] to NNs (concretizing thoughts in Werbos' 1974 thesis).*

[BP4] J. Schmidhuber (AI Blog, 2014; updated 2020). Who invented backpropagation? More. [DL2]

[BP5] A. Griewank (2012). Who invented the reverse mode of differentiation? Documenta Mathematica, Extra Volume ISMP (2012): 389-400.

[BPTT1] P. J. Werbos. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE 78.10, 1550-1560, 1990.

[BPTT2] R. J. Williams and D. Zipser. Gradient-based learning algorithms for recurrent networks. In: Backpropagation: Theory, architectures, and applications, p 433, 1995.

[BPTT3] B. A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. IEEE Transactions on Neural Networks, 6(5):1212-1228, 1995.

[BUR18] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, A. A. Efros. Large-scale study of curiosity-driven learning. Preprint arXiv:1808.04355, 2018.

[BRE] H. J. Bremermann (1982). Minimum energy requirements of information transfer and computing, International Journal of Theoretical Physics, 21, 203-217, 1982.

[BRI] Bridle, J.S. (1990). Alpha-Nets: A Recurrent "Neural" Network Architecture with a Hidden Markov Model Interpretation, Speech Communication, vol. 9, no. 1, pp. 83-92.

[BRU1] H. Bruderer. Computing history beyond the UK and US: selected landmarks from continental Europe. Communications of the ACM 60.2 (2017): 76-84.

[BRU2] H. Bruderer. Meilensteine der Rechentechnik. 2 volumes, 3rd edition. Walter de Gruyter GmbH & Co KG, 2020.

[BRU3] H. Bruderer. Milestones in Analog and Digital Computing. 2 volumes, 3rd edition. Springer Nature Switzerland AG, 2020.

[BRU4] H. Bruderer. The Birthplace of Artificial Intelligence? Communications of the ACM, BLOG@CACM, Nov 2017. Link.

[BRO21] D. C. Brock (2021). Cybernetics, Computer Design, and a Meeting of the Minds. An influential 1951 conference in Paris considered the computer as a model of—and for—the human mind. IEEE Spectrum, 2021. Link.

[BW] H. Bourlard, C. J. Wellekens (1989). Links between Markov models and multilayer perceptrons. NIPS 1989, p. 502-510.

[CAN] G. Cantor (1891). Ueber eine elementare Frage der Mannigfaltigkeitslehre. Jahresbericht der Deutschen Mathematiker-Vereinigung, 1:75-78. *English translation: W. B. Ewald (ed.). From Immanuel Kant to David Hilbert: A Source Book in the Foundations of Mathematics, Volume 2, pp. 920-922. Oxford University Press, 1996.*

[CDI] G. E. Hinton. Training products of experts by minimizing contrastive divergence. Neural computation 14.8 (2002): 1771-1800.

[CHO15] F. Chollet (2015). Keras.

[CHU] A. Church (1935). An unsolvable problem of elementary number theory. Bulletin of the American Mathematical Society, 41: 332-333. Abstract of a talk given on 19 April 1935, to the American Mathematical Society. Also in American Journal of Mathematics, 58(2), 345-363 (1 Apr 1936). *First explicit proof that the Entscheidungsproblem (decision problem) does not have a general solution.*

[CNN1] K. Fukushima: Neural network model for a mechanism of pattern recognition unaffected by shift in position—Neocognitron. Trans. IECE, vol. J62-A, no. 10, pp. 658-665, 1979. *The first deep convolutional neural network architecture, with alternating convolutional layers and downsampling layers. In Japanese. English version: [CNN1+]. More in Scholarpedia.*

[CNN1+] K. Fukushima: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, vol. 36, no. 4, pp. 193-202 (April 1980). Link.

[CNN1a] A. Waibel. Phoneme Recognition Using Time-Delay Neural Networks. Meeting of IEICE, Tokyo, Japan, 1987. *First application of backpropagation[BP1-5] and weight-sharing to a convolutional architecture.*

[CNN1b] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang. Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 3, pp. 328-339, March 1989. Based on [CNN1a].

[CNN1c] Bower Award Ceremony 2021: Jürgen Schmidhuber lauds Kunihiko Fukushima. YouTube video, 2021.

[CNN2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, 1989. PDF.

[CNN3a] K. Yamaguchi, K. Sakamoto, A. Kenji, T. Akabane, Y. Fujimoto. A Neural Network for Speaker-Independent Isolated Word Recognition. First International Conference on Spoken Language Processing (ICSLP 90), Kobe, Japan, Nov 1990. *An NN with convolutions using Max-Pooling instead of Fukushima's Spatial Averaging.[CNN1]*

[CNN3] Weng, J., Ahuja, N., and Huang, T. S. (1993). Learning recognition and segmentation of 3-D objects from 2-D images. Proc. 4th Intl. Conf. Computer Vision, Berlin, Germany, pp.

121-128. *A CNN whose downsampling layers use Max-Pooling (which has become very popular) instead of Fukushima's Spatial Averaging.*[CNN1]

[CNN4] M. A. Ranzato, Y. LeCun: A Sparse and Locally Shift Invariant Feature Extractor Applied to Document Images. Proc. ICDAR, 2007

[CNN5a] S. Behnke. Learning iterative image reconstruction in the neural abstraction pyramid. International Journal of Computational Intelligence and Applications, 1(4):427-438, 1999.

[CNN5b] S. Behnke. Hierarchical Neural Networks for Image Interpretation, volume LNCS 2766 of Lecture Notes in Computer Science. Springer, 2003.

[CNN5c] D. Scherer, A. Mueller, S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In Proc. International Conference on Artificial Neural Networks (ICANN), pages 92-101, 2010.

[CO1] J. Koutnik, F. Gomez, J. Schmidhuber (2010). Evolving Neural Networks in Compressed Weight Space. *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO-2010), Portland, 2010. PDF.

[CO2] J. Koutnik, G. Cuccu, J. Schmidhuber, F. Gomez. Evolving Large-Scale Neural Networks for Vision-Based Reinforcement Learning. *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO), Amsterdam, July 2013. PDF. *The first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning, without any unsupervised pre-training.*

[CO3] R. K. Srivastava, J. Schmidhuber, F. Gomez. Generalized Compressed Network Search. Proc. GECCO 2012. PDF.

[CON16] J. Carmichael (2016). Artificial Intelligence Gained Consciousness in 1991. Why A.I. pioneer Jürgen Schmidhuber is convinced the ultimate breakthrough already happened. Inverse, 2016. Link.

[CONN21] Since November 2021: Comments on version 1 of the report[T22] in the Connectionists Mailing List, perhaps the oldest mailing list on artificial neural networks. Link to the archive.

[CTC] A. Graves, S. Fernandez, F. Gomez, J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. ICML 06, Pittsburgh, 2006. PDF.

[CUB0] R. J. Williams. Complexity of exact gradient computation algorithms for recurrent neural networks. Technical Report NU-CCS-89-27, Northeastern University, College of Computer Science, 1989.

[CUB2] 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. PDF.

[CW] J. Koutnik, K. Greff, F. Gomez, J. Schmidhuber. A Clockwork RNN. Proc. 31st International Conference on Machine Learning (ICML), p. 1845-1853, Beijing, 2014. Preprint arXiv:1402.3511 [cs.NE].

[DAN] J. Schmidhuber (AI Blog, 2021). 10-year anniversary. In 2011, DanNet triggered the deep convolutional neural network (CNN) revolution. *Named after Schmidhuber's outstanding postdoc Dan Ciresan, it was the first deep and fast CNN to win international computer vision contests, and had a temporary monopoly on winning them, driven by a very fast implementation based on graphics processing units (GPUs). 1st superhuman result in 2011.* [DAN1] *Now everybody is using this approach.*

[DAN1] J. Schmidhuber (AI Blog, 2011; updated 2021 for 10th birthday of DanNet): First superhuman visual pattern recognition. *At the IJCNN 2011 computer vision competition in Silicon Valley, the artificial neural network called DanNet performed twice better than humans, three times better than the closest artificial competitor (from LeCun's team), and six times better than the best non-neural method.*

[DEC] J. Schmidhuber (AI Blog, 02/20/2020, updated 2021, 2022). The 2010s: Our Decade of Deep Learning / Outlook on the 2020s. *The recent decade's most important developments and industrial applications based on the AI of Schmidhuber's team, with an outlook on the 2020s, also addressing privacy and data markets.*

[DEEP1] Ivakhnenko, A. G. and Lapa, V. G. (1965). Cybernetic Predicting Devices. CCM Information Corporation. *First working Deep Learners with many layers, learning internal representations.*

[DEEP1a] Ivakhnenko, Alexey Grigorevich. The group method of data of handling; a rival of the method of stochastic approximation. Soviet Automatic Control 13 (1968): 43-55.

[DEEP2] Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. IEEE Transactions on Systems, Man and Cybernetics, (4):364-378.

[DIST2] O. Vinyals, J. A. Dean, G. E. Hinton. Distilling the Knowledge in a Neural Network. Preprint arXiv:1503.02531 [stat.ML], 2015. *The authors did not cite Schmidhuber's original 1991 NN distillation procedure,* [UN0-2][MIR](Sec. 2) *not even in the later patent application US20150356461A1.*

[DL1] J. Schmidhuber, 2015. Deep learning in neural networks: An overview. Neural Networks, 61, 85-117. More. *Got the first Best Paper Award ever issued by the journal Neural Networks, founded in 1988.*

[DL2] J. Schmidhuber, 2015. Deep Learning. Scholarpedia, 10(11):32832.

[DL3] Y. LeCun, Y. Bengio, G. Hinton (2015). Deep Learning. Nature 521, 436-444. HTML. *A "survey" of deep learning that does not mention the pioneering works of deep learning [T22].*

[DL3a] Y. Bengio, Y. LeCun, G. Hinton (2021). Turing Lecture: Deep Learning for AI. Communications of the ACM, July 2021. HTML. Local copy (HTML only). *Another "survey" of deep learning that does not mention the pioneering works of deep learning [T22].*

[DL4] J. Schmidhuber (AI Blog, 2017). Our impact on the world's most valuable public companies: Apple, Google, Microsoft, Facebook, Amazon... *By 2015-17, neural nets developed in Schmidhuber's labs were on over 3 billion devices such as smartphones, and used many billions of times per day, consuming a significant fraction of the world's compute. Examples: greatly improved (CTC-based) speech recognition on all Android phones, greatly improved machine translation through Google Translate and Facebook (over 4 billion LSTM-based translations per day), Apple's Siri and Quicktype on all iPhones, the answers of Amazon's Alexa, etc. Google's 2019 on-device speech recognition (on the phone, not the server) is still based on LSTM.*

[DL6] F. Gomez and J. Schmidhuber. Co-evolving recurrent neurons learn deep memory POMDPs. In *Proc. GECCO'05*, Washington, D. C., pp. 1795-1802, ACM Press, New York, NY, USA, 2005. PDF.

[DL6a] J. Schmidhuber (AI Blog, Nov 2020). 15-year anniversary: 1st paper with "learn deep" in the title (2005). *The deep reinforcement learning & neuroevolution developed in Schmidhuber's lab solved problems of depth 1000 and more.[DL6] Soon after its publication, everybody started talking about "deep learning." Causality or correlation?*

[DL7] "Deep Learning ... moving beyond shallow machine learning since 2006!" Web site deeplearning.net of Y. Bengio's MILA (2015, retrieved May 2020; compare the version in the Internet Archive), referring to Hinton's[UN4] and Bengio's[UN5] *unsupervised* pre-training for deep NNs[UN4] (2006) although this type of deep learning dates back to Schmidhuber's work of 1991. [UN1-2][UN]

[DLC] J. Schmidhuber (AI Blog, June 2015). Critique of Paper by self-proclaimed[DLC2] "Deep Learning Conspiracy" (Nature 521 p 436). *The inventor of an important method should get credit for inventing it. She may not always be the one who popularizes it. Then the popularizer should get credit for popularizing it (but not for inventing it). More on this under [T22].*

[DLC1] Y. LeCun. IEEE Spectrum Interview by L. Gomes, Feb 2015. *Quote: "A lot of us involved in the resurgence of Deep Learning in the mid-2000s, including Geoff Hinton, Yoshua Bengio, and myself—the so-called 'Deep Learning conspiracy' ..."*

[DLC2] M. Bergen, K. Wagner (2015). Welcome to the AI Conspiracy: The 'Canadian Mafia' Behind Tech's Latest Craze. Vox recode, 15 July 2015. *Quote: "... referred to themselves as the 'deep learning conspiracy.' Others called them the 'Canadian Mafia.'"*

[DLH] J. Schmidhuber (AI Blog, 2022). Annotated History of Modern AI and Deep Learning. Technical Report IDSIA-22-22, IDSIA, Lugano, Switzerland, 2022. Preprint arXiv:2212.11279. Tweet of 2022.

[DM1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller. Playing Atari with Deep Reinforcement Learning. Tech Report, 19 Dec. 2013, arxiv:1312.5602.

[DM2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis. Human-level control through deep reinforcement learning. Nature, vol. 518, p 1529, 26 Feb. 2015. Link. *DeepMind's first famous* *paper. Its abstract claims: "While reinforcement learning agents have achieved some successes in a variety of domains, their applicability has previously been limited to domains in which useful features can be handcrafted, or to domains with fully observed, low-dimensional state spaces." It also claims to bridge "the divide between high-dimensional sensory inputs and actions." Similarly, the first sentence of the abstract of the earlier tech report version[DM1] of [DM2] claims to "present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning." However, the first such system (requiring no unsupervised pre-training) was created earlier by Jan Koutnik et al. in Schmidhuber's lab.[CO2] DeepMind was co-founded by Shane Legg, a PhD student from this lab; he and Daan Wierstra (another PhD student of Schmidhuber and DeepMind's 1st employee) were the first persons at DeepMind who had AI publications and PhDs in computer science. More.*

[DM2a] D. Silver et al. A general reinforcement learning algorithm that masters chess, Shogi, and Go through self-play. Science 362.6419:1140-1144, 2018.

[DM3] S. Stanford. DeepMind's AI, AlphaStar Showcases Significant Progress Towards AGI. Medium ML Memoirs, 2019. *Alphastar has a "deep LSTM core."*

[DM4] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli & D. Hassabis. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583-589, 2021. *DeepMind's breakthrough application of deep learning did not cite Hochreiter et al.'s first successful application [HO07] of deep learning to protein folding (2007).*

[DIF1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022. Preprint arXiv:2112.10752, LMU Munich, 2021.

[DIF2] C. Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. Physical Review E, 1997.

[DIF3] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. CoRR, abs/1503.03585, 2015.

[DIF4] O. Ronneberger, P. Fischer, T. Brox. Unet: Convolutional networks for biomedical image segmentation. In MICCAI (3), vol. 9351 of Lecture Notes in Computer Science, pages 234-241. Springer, 2015.

[DIF5] J. Ho, A. Jain, P. Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems 33:6840-6851, 2020.

[DNC] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwinska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, D. Hassabis.

Hybrid computing using a neural network with dynamic external memory. Nature, 538:7626, p 471, 2016. *This work of DeepMind did not cite the original work of the early 1990s on neural networks learning to control dynamic external memories.*[PDA1-2][FWP0-1]

[Drop1] S. J. Hanson (1990). A Stochastic Version of the Delta Rule, PHYSICA D,42, 265-272. *What's now called "dropout" is a variation of the stochastic delta rule—compare preprint arXiv:1808.03578, 2018.*

[Drop2] N. Frazier-Logue, S. J. Hanson (2020). The Stochastic Delta Rule: Faster and More Accurate Deep Learning Through Adaptive Weight Noise. Neural Computation 32(5):1018-1032.

[Drop3] J. Hertz, A. Krogh, R. Palmer (1991). Introduction to the Theory of Neural Computation. Redwood City, California: Addison-Wesley Pub. Co., pp. 45-46.

[Drop4] N. Frazier-Logue, S. J. Hanson (2018). Dropout is a special case of the stochastic delta rule: faster and more accurate deep learning. Preprint arXiv:1808.03578, 2018.

[DYNA90] R. S. Sutton (1990). Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. Machine Learning Proceedings 1990, of the Seventh International Conference, Austin, Texas, June 21-23, 1990, p 216-224.

[DYNA91] R. S. Sutton (1991). Dyna, an integrated architecture for learning, planning, and reacting. ACM Sigart Bulletin 2.4 (1991):160-163.

[ELM1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: A new learning scheme of feedforward neural networks. Proc. IEEE Int. Joint Conf. on Neural Networks, Vol. 2, 2004, pp. 985-990. *This paper does not mention that the "ELM" concept goes back to Rosenblatt's work in the 1950s.*[R62][T22]

[ELM2] ELM-ORIGIN, 2004. The Official Homepage on Origins of Extreme Learning Machines (ELM). "Extreme Learning Machine Duplicates Others' Papers from 1988-2007." Local copy. *This overview does not mention that the "ELM" concept goes back to Rosenblatt's work in the 1950s.*[R62][T22]

[ENS1] R. E. Schapire. The strength of weak learnability. Machine Learning, 5:197-227, 1990.

[ENS2] D. H. Wolpert. Stacked generalization. Neural Networks, 5(2):241-259, 1992.

[ENS3] L. Breiman. Bagging predictors. Machine Learning, 24:123-140, 1996.

[ENS4] T. G. Dietterich. Ensemble methods in machine learning. In Multiple classifier systems, pages 1-15. Springer, 2000.

[EVO1] N. A. Barricelli. Esempi numerici di processi di evoluzione. Methodos: 45-68, 1954. *Possibly the first publication on artificial evolution.*

[EVO2] L. Fogel, A. Owens, M. Walsh. Artificial Intelligence through Simulated Evolution. Wiley, New York, 1966.

[EVO3] I. Rechenberg. Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dissertation, 1971.

[EVO4] H. P. Schwefel. Numerische Optimierung von Computer-Modellen. Dissertation, 1974.

[EVO5] J. H. Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.

[EVO6] S. F. Smith. A Learning System Based on Genetic Adaptive Algorithms, PhD Thesis, Univ. Pittsburgh, 1980

[EVO7] N. L. Cramer. A representation for the adaptive generation of simple sequential programs. In J. J. Grefenstette, editor, Proceedings of an International Conference on Genetic Algorithms and Their Applications, Carnegie-Mellon University, July 24-26, 1985, Hillsdale NJ, 1985. Lawrence Erlbaum Associates.

[EVONN1] Montana, D. J. and Davis, L. (1989). Training feedforward neural networks using genetic algorithms. In Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)—Volume 1, IJCAI'89, pages 762â€“767, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[EVONN2] Miller, G., Todd, P., and Hedge, S. (1989). Designing neural networks using genetic algorithms. In Proceedings of the 3rd International Conference on Genetic Algorithms, pages 379â€“384. Morgan Kauffman.

[EVONN3] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. Complex Systems, 4:461-476, 1990.

[FA15] Intelligente Roboter werden vom Leben fasziniert sein. (Intelligent robots will be fascinated by life.) FAZ, 1 Dec 2015. Link.

[FAKE] H. Hopf, A. Krief, G. Mehta, S. A. Matlin. Fake science and the knowledge crisis: ignorance can be fatal. Royal Society Open Science, May 2019. *Quote: "Scientists must be willing to speak out when they see false information being presented in social media, traditional print or broadcast press" and "must speak out against false information and fake science in circulation and forcefully contradict public figures who promote it."*

[FAKE2] L. Stenflo. Intelligent plagiarists are the most dangerous. Nature, vol. 427, p. 777 (Feb 2004). *Quote: "What is worse, in my opinion, ..., are cases where scientists rewrite previous findings in different words, purposely hiding the sources of their ideas, and then during subsequent years forcefully claim that they have discovered new phenomena.*

[FAST] C. v.d. Malsburg. Tech Report 81-2, Abteilung f. Neurobiologie, Max-Planck Institut f. Biophysik und Chemie, Goettingen, 1981. *First paper on fast weights or dynamic links.*

[FASTa] J. A. Feldman. Dynamic connections in neural networks. Biological Cybernetics, 46(1):27-39, 1982. *2nd paper on fast weights.*

[FASTb] G. E. Hinton, D. C. Plaut. Using fast weights to deblur old memories. Proc. 9th annual conference of the Cognitive Science Society (pp. 177-186), 1987. *3rd paper on fast weights (two types of weights with different learning rates).*

[FB17] By 2017, Facebook used LSTM to handle over 4 billion automatic translations per day (The Verge, August 4, 2017); see also Facebook blog by J.M. Pino, A. Sidorov, N.F. Ayan (August 3, 2017)

[FDL] J. Schmidhuber (AI Blog, 2013). My First Deep Learning System of 1991 + Deep Learning Timeline 1960-2013.

[FEI63] E. A. Feigenbaum, J. Feldman. Computers and thought. McGraw-Hill: New York, 1963.

[FEI83] E. A. Feigenbaum, P. McCorduck. The fifth generation. Addison-Wesley Publishers, 1983.

[FI22] R. A. Fisher. On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character; 222.594-604:309-368, 1922.

[FM] S. Hochreiter and J. Schmidhuber. Flat minimum search finds simple nets. Technical Report FKI-200-94, Fakultät für Informatik, Technische Universität München, December 1994. PDF.

[FRE] G. Frege (1879). Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle an der Saale: Verlag Louis Nebert. *The first formal language / formal proofs—basis of modern logic and programming languages.*

[FU77] K. S. Fu. Syntactic Pattern Recognition and Applications. Berlin, Springer, 1977.

[FWP] J. Schmidhuber (AI Blog, 26 March 2021, updated 2022). 26 March 1991: Neural nets learn to program neural nets with fast weights—like Transformer variants. 2021: New stuff! *30-year anniversary of a now popular alternative[FWP0-1] to recurrent NNs. A slow feedforward NN learns by gradient descent to program the changes of the fast weights[FAST,FASTa,b] of another NN, separating memory and control like in traditional computers. Such Fast Weight Programmers[FWP0-6,FWPMETA1-8] can learn to memorize past data, e.g., by computing fast weight changes through additive outer products of self-invented activation patterns[FWP0-1] (now often called keys and values for self-attention[TR1-6]). The similar Transformers[TR1-2] combine this with projections and softmax and are now widely used in natural language processing. For long input sequences, their efficiency was improved through Transformers with linearized self-attention[TR5-6] which are formally equivalent to Schmidhuber's 1991 outer product-based Fast Weight Programmers (apart from normalization). In 1993, he introduced the attention terminology[FWP2] now used in this context,[ATT] and extended the approach to RNNs that program themselves. See tweet of 2022.*

[FWP0] J. Schmidhuber. Learning to control fast-weight memories: An alternative to recurrent nets. Technical Report FKI-147-91, Institut für Informatik, Technische Universität München, 26 March 1991. PDF. *First paper on fast weight programmers that separate storage and control: a slow net learns by gradient descent to compute weight changes of a fast net. The outer*

product-based version (Eq. 5) is now known as a "Transformer with linearized self-attention" (apart from normalization).[FWP]

[FWP1] J. Schmidhuber. Learning to control fast-weight memories: An alternative to recurrent nets. Neural Computation, 4(1):131-139, 1992. Based on [FWP0]. PDF. HTML. Pictures (German). See tweet of 2022 for 30-year anniversary.

[FWP2] J. Schmidhuber. Reducing the ratio between learning complexity and number of time-varying variables in fully recurrent nets. In Proceedings of the International Conference on Artificial Neural Networks, Amsterdam, pages 460-463. Springer, 1993. PDF. *First recurrent NN-based fast weight programmer using outer products, introducing the terminology of learning "internal spotlights of attention."*

[FWP3] I. Schlag, J. Schmidhuber. Gated Fast Weights for On-The-Fly Neural Program Generation. Workshop on Meta-Learning, @N(eur)IPS 2017, Long Beach, CA, USA.

[FWP3a] I. Schlag, J. Schmidhuber. Learning to Reason with Third Order Tensor Products. Advances in Neural Information Processing Systems (N(eur)IPS), Montreal, 2018. Preprint: arXiv:1811.12143. PDF.

[FWP4a] J. Ba, G. Hinton, V. Mnih, J. Z. Leibo, C. Ionescu. Using Fast Weights to Attend to the Recent Past. NIPS 2016. PDF. *Very similar to [FWP0-2], in both motivation [FWP2] and execution.*

[FWP4b] D. Bahdanau, K. Cho, Y. Bengio (2014). Neural Machine Translation by Jointly Learning to Align and Translate. Preprint arXiv:1409.0473 [cs.CL]. *This work on "attention" did not cite Schmidhuber's much earlier original work of 1991-1993 on soft attention and Transformers with linearized self-attention.*[FWP,FWP0-2,6][ATT]

[FWP4d] Y. Tang, D. Nguyen, D. Ha (2020). Neuroevolution of Self-Interpretable Agents. Preprint: arXiv:2003.08165.

[FWP5] F. J. Gomez and J. Schmidhuber. Evolving modular fast-weight networks for control. In W. Duch et al. (Eds.): *Proc. ICANN'05,* LNCS 3697, pp. 383-389, Springer-Verlag Berlin Heidelberg, 2005. PDF. HTML overview. *Reinforcement-learning fast weight programmer.*

[FWP6] I. Schlag, K. Irie, J. Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers. ICML 2021. Preprint: arXiv:2102.11174.

[FWP7] K. Irie, I. Schlag, R. Csordas, J. Schmidhuber. Going Beyond Linear Transformers with Recurrent Fast Weight Programmers. Preprint: arXiv:2106.06295 (June 2021).

[FWPMETA1] J. Schmidhuber. Steps towards `self-referential' learning. Technical Report CU-CS-627-92, Dept. of Comp. Sci., University of Colorado at Boulder, November 1992. *First recurrent NN-based fast weight programmer that can learn to run a learning algorithm or weight change algorithm on itself.*

[FWPMETA2] J. Schmidhuber. A self-referential weight matrix. In *Proceedings of the International Conference on Artificial Neural Networks, Amsterdam*, pages 446-451. Springer,

1993. PDF.

[FWPMETA3] J. Schmidhuber. An introspective network that can learn to run its own weight change algorithm. In *Proc. of the Intl. Conf. on Artificial Neural Networks, Brighton*, pages 191-195. IEE, 1993.

[FWPMETA4] J. Schmidhuber. A neural network that embeds its own meta-levels. In *Proc. of the International Conference on Neural Networks '93, San Francisco*. IEEE, 1993.

[FWPMETA5] J. Schmidhuber. Habilitation thesis, TUM, 1993. PDF. *A recurrent neural net with a self-referential, self-reading, self-modifying weight matrix can be found here.*

[FWPMETA6] L. Kirsch and J. Schmidhuber. Meta Learning Backpropagation & Improving It. Advances in Neural Information Processing Systems (NeurIPS), 2021. Preprint arXiv:2012.14905 [cs.LG], 2020.

[FWPMETA7] I. Schlag, T. Munkhdalai, J. Schmidhuber. Learning Associative Inference Using Fast Weight Memory. ICLR 2021. Report arXiv:2011.07831 [cs.AI], 2020.

[FWPMETA8] K. Irie, I. Schlag, R. Csordas, J. Schmidhuber. A Modern Self-Referential Weight Matrix That Learns to Modify Itself. International Conference on Machine Learning (ICML), 2022. Preprint: arXiv:2202.05780.

[FWPMETA9] L. Kirsch and J. Schmidhuber. Self-Referential Meta Learning. First Conference on Automated Machine Learning (Late-Breaking Workshop), 2022.

[GAU09] C. F. Gauss. Theoria motus corporum coelestium in sectionibus conicis solem ambientium, 1809.

[GAU21] C. F. Gauss. Theoria combinationis observationum erroribus minimis obnoxiae (Theory of the combination of observations least subject to error), 1821.

[G63] R. J Glauber (1963). Time-dependent statistics of the Ising model. Journal of Mathematical Physics, 4(2):294-307, 1963.

[GD'] C. Lemarechal. Cauchy and the Gradient Method. Doc Math Extra, pp. 251-254, 2012.

[GD''] J. Hadamard. Memoire sur le probleme d'analyse relatif a Vequilibre des plaques elastiques encastrees. Memoires presentes par divers savants estrangers Ã l'Academie des Sciences de l'Institut de France, 33, 1908.

[GDa] Y. Z. Tsypkin (1966). Adaptation, training and self-organization automatic control systems, Avtomatika I Telemekhanika, 27, 23-61. *On gradient descent-based on-line learning for non-linear systems.*

[GDb] Y. Z. Tsypkin (1971). Adaptation and Learning in Automatic Systems, Academic Press, 1971. *On gradient descent-based on-line learning for non-linear systems.*

[GD1] S. I. Amari (1967). A theory of adaptive pattern classifier, IEEE Trans, EC-16, 279-307 (Japanese version published in 1965). PDF. *Probably the first paper on using stochastic gradient descent*[STO51-52] *for learning in multilayer neural networks (without specifying the specific gradient descent method now known as reverse mode of automatic differentiation or backpropagation*[BP1]*).*

[GD2] S. I. Amari (1968). Information Theory—Geometric Theory of Information, Kyoritsu Publ., 1968 (in Japanese). OCR-based PDF scan of pages 94-135 (see pages 119-120). *Contains computer simulation results for a five layer network (with 2 modifiable layers) which learns internal representations to classify non-linearily separable pattern classes.*

[GD2a] H. Saito (1967). Master's thesis, Graduate School of Engineering, Kyushu University, Japan. *Implementation of Amari's 1967 stochastic gradient descent method for multilayer perceptrons.*[GD1] *(S. Amari, personal communication, 2021.)*

[GD3] S. I. Amari (1977). Neural Theory of Association and Concept Formation. Biological Cybernetics, vol. 26, p. 175-185, 1977. *See Section 3.1 on using gradient descent for learning in multilayer networks.*

[GGP] F. Faccio, V. Herrmann, A. Ramesh, L. Kirsch, J. Schmidhuber. Goal-Conditioned Generators of Deep Policies. Preprint arXiv/2207.01570, 4 July 2022 (submitted in May 2022).

[GLA85] W. Glasser. Control theory. New York: Harper and Row, 1985.

[GM3] J. Schmidhuber (2003). Gödel Machines: Self-Referential Universal Problem Solvers Making Provably Optimal Self-Improvements. Preprint arXiv:cs/0309048 (2003). More.

[GM6] J. Schmidhuber (2006). Gödel machines: Fully Self-Referential Optimal Universal Self-Improvers. In B. Goertzel and C. Pennachin, eds.: *Artificial General Intelligence,* p. 199-226, 2006. PDF.

[GM9] J. Schmidhuber (2009). Ultimate Cognition *à la* Gödel. *Cognitive Computation* 1(2):177-193, 2009. PDF. More.

[GSR] H. Sak, A. Senior, K. Rao, F. Beaufays, J. Schalkwyk—Google Speech Team. Google voice search: faster and more accurate. Google Research Blog, Sep 2015, see also Aug 2015 Google's speech recognition based on CTC and LSTM.

[GSR15] Dramatic improvement of Google's speech recognition through LSTM: Alphr Technology, Jul 2015, or 9to5google, Jul 2015

[GSR19] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. Chai Sim, T. Bagby, S. Chang, K. Rao, A. Gruenstein. Streaming end-to-end speech recognition for mobile devices. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[GT16] Google's dramatically improved Google Translate of 2016 is based on LSTM, e.g., WIRED, Sep 2016, or siliconANGLE, Sep 2016

[GAN0] O. Niemitalo. A method for training artificial neural networks to generate missing data within a variable context. Blog post, Internet Archive, 2010. *A blog post describing basic ideas*[AC][AC90,AC90b][AC20] *of GANs.*

[GAN1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial nets. NIPS 2014, 2672-2680, Dec 2014. *A description of GANs that does not cite Schmidhuber's original GAN principle of 1990*[AC][AC90,AC90b][AC20][R2][T22] *(also containing wrong claims about Schmidhuber's adversarial NNs for Predictability Minimization*[PM0-2][AC20][T22]*).*

[GAN2] T. Karras, S. Laine, T. Aila. A style-based generator architecture for generative adversarial networks. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 4401-4410, 2019.

[GOD] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik, 38:173-198, 1931. *In the early 1930s, Gödel founded theoretical computer science. He identified fundamental limits of mathematics, theorem proving, computing, and Artificial Intelligence.*

[GOD34] K. Gödel (1934). On undecidable propositions of formal mathematical systems. Notes by S. C. Kleene and J. B. Rosser on lectures at the Institute for Advanced Study, Princeton, New Jersey, 1934, 30 pp. (Reprinted in M. Davis, (ed.), The Undecidable. Basic Papers on Undecidable Propositions, Unsolvable Problems, and Computable Functions, Raven Press, Hewlett, New York, 1965.) *Gödel introduced a universal coding language.*

[GOD56] R. J. Lipton and K. W. Regan. Gödel's lost letter and P=NP. Link. *Gödel identified what's now the most famous open problem of computer science.*

[GOD86] K. Gödel. Collected works Volume I: Publications 1929-36, S. Feferman et. al., editors, Oxford Univ. Press, Oxford, 1986.

[GOD10] V. C. Nadkarni. Gödel, Einstein and proof for God. The Economic Times, 2010.

[GOD21] J. Schmidhuber (2021). 90th anniversary celebrations: 1931: Kurt Gödel, founder of theoretical computer science, shows limits of math, logic, computing, and artificial intelligence. *This was number 1 on Hacker News.*

[GOD21a] J. Schmidhuber (2021). Als Kurt Gödel die Grenzen des Berechenbaren entdeckte. (When Kurt Gödel discovered the limits of computability.) Frankfurter Allgemeine Zeitung, 16/6/2021.

[GOD21b] J. Schmidhuber (AI Blog, 2021). 80. Jahrestag: 1931: Kurt Gödel, Vater der theoretischen Informatik, entdeckt die Grenzen des Berechenbaren und der künstlichen Intelligenz.

[GOL] C. Goller & A. Küchler (1996). Learning task-dependent distributed representations by backpropagation through structure. Proceedings of International Conference on Neural Networks (ICNN'96). Vol. 1, p. 347-352 IEEE, 1996. Based on TR AR-95-02, TU Munich, 1995.

[GPT3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. Language Models are Few-Shot Learners (2020). Preprint arXiv/2005.14165.

[GPUNN] Oh, K.-S. and Jung, K. (2004). GPU implementation of neural networks. Pattern Recognition, 37(6):1311-1314. *Speeding up traditional NNs on GPU by a factor of 20.*

[GPUCNN] K. Chellapilla, S. Puri, P. Simard. High performance convolutional neural networks for document processing. International Workshop on Frontiers in Handwriting Recognition, 2006. *Speeding up shallow CNNs on GPU by a factor of 4.*

[GPUCNN1] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber. Flexible, High Performance Convolutional Neural Networks for Image Classification. *International Joint Conference on Artificial Intelligence (IJCAI-2011, Barcelona)*, 2011. PDF. ArXiv preprint. *Speeding up deep CNNs on GPU by a factor of 60. Used to win four important computer vision competitions 2011-2012 before others won any with similar approaches.*

[GPUCNN2] D. C. Ciresan, U. Meier, J. Masci, J. Schmidhuber. A Committee of Neural Networks for Traffic Sign Classification. *International Joint Conference on Neural Networks (IJCNN-2011, San Francisco)*, 2011. PDF. HTML overview. *First superhuman performance in a computer vision contest, with half the error rate of humans, and one third the error rate of the closest competitor.[DAN1] This led to massive interest from industry.*

[GPUCNN3] D. C. Ciresan, U. Meier, J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification. Proc. *IEEE Conf. on Computer Vision and Pattern Recognition CVPR 2012*, p 3642-3649, July 2012. PDF. Longer TR of Feb 2012: arXiv:1202.2745v1 [cs.CV]. More.

[GPUCNN3a] D. Ciresan, A. Giusti, L. Gambardella, J. Schmidhuber. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. NIPS 2012, Lake Tahoe, 2012.

[GPUCNN4] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS 25, MIT Press, Dec 2012. PDF. *This paper describes AlexNet, which is similar to the earlier DanNet,[DAN,DAN1][R6] the first pure deep CNN to win computer vision contests in 2011[GPUCNN2-3,5] (AlexNet and VGG Net[GPUCNN9] followed in 2012-2014). [GPUCNN4] emphasizes benefits of Fukushima's ReLUs (1969)[RELU1] and dropout (a variant of Hanson 1990 stochastic delta rule)[Drop1-4] but neither cites the original work[RELU1][Drop1] nor the basic CNN architecture (Fukushima, 1979).[CNN1]*

[GPUCNN5] J. Schmidhuber (AI Blog, 2017; updated 2021 for 10th birthday of DanNet): History of computer vision contests won by deep CNNs since 2011. DanNet was the first CNN to win one, and won 4 of them in a row before the similar AlexNet/VGG Net and the Resnet (a Highway Net with open gates) joined the party. Today, deep CNNs are standard in computer vision.

[GPUCNN6] J. Schmidhuber, D. Ciresan, U. Meier, J. Masci, A. Graves. On Fast Deep Nets for AGI Vision. In Proc. Fourth Conference on Artificial General Intelligence (AGI-11), Google, Mountain View, California, 2011. PDF.

[GPUCNN7] D. C. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber. Mitosis Detection in Breast Cancer Histology Images using Deep Neural Networks. MICCAI 2013. PDF.

[GPUCNN8] J. Schmidhuber (AI Blog, 2017; updated 2021 for 10th birthday of DanNet). First deep learner to win a contest on object detection in large images— first deep learner to win a medical imaging contest (2012). Link. *How the Swiss AI Lab IDSIA used GPU-based CNNs to win the ICPR 2012 Contest on Mitosis Detection and the MICCAI 2013 Grand Challenge.*

[GPUCNN9] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. Preprint arXiv:1409.1556 (2014).

[GRO69] S. Grossberg. Some networks that can learn, remember, and reproduce any number of complicated space-time patterns, Indiana University Journal of Mathematics and Mechanics, 19:53-91, 1969.

[H86] J. L. van Hemmen (1986). Spin-glass models of a neural network. Phys. Rev. A 34, 3435, 1 Oct 1986.

[H88] H. Sompolinsky (1988). Statistical Mechanics of Neural Networks. Physics Today 41, 12, 70, 1988.

[H90] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. Physica D: Nonlinear Phenomena, 42(1-3):228-234, 1990.

[HAB1] V. Smil. Detonator of the population explosion. Nature, July 29 1999, p 415.

[HAB2] J. Schmidhuber (Blog, 2000). Most influential persons of the 20th century (according to Nature, 1999). *The Haber-Bosch process has often been called the most important invention of the 20th century[HAB1] as it "detonated the population explosion," driving the world's population from 1.6 billion in 1900 to about 8 billion today. Without it, half of humanity would perish. Billions of people would never have existed without it; soon it will sustain 2 out of 3 persons.*

[HAI14] T. Haigh (2014). Historical reflections. Actually, Turing did not invent the computer. Communications of the ACM, Vol. 57(1): 36-41, Jan 2014. PDF.

[HAI14b] T. Haigh, M. Priestley, C. Rope (2014). Reconsidering the Stored-Program Concept. IEEE Annals of the History of Computing. IEEE, 2014. PDF.

[HB96] S. El Hihi, Y. Bengio. Hierarchical recurrent neural networks for long-term dependencies. NIPS, 1996. *Bengio claimed[YB20] that in 1995 he "introduced the use of a hierarchy of time scales to combat the vanishing gradients issue" although Schmidhuber's publications on exactly this topic date back to 1991-93.[UN0-2][UN]*

[HEB49] D. O. Hebb. The Organization of Behavior. Wiley, New York, 1949.

[HEL] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. Neural Computation, 7:889-904, 1995. *An unsupervised learning algorithm related to Schmidhuber's supervised Neural Heat Exchanger.*[NHE]

[HIN] J. Schmidhuber (AI Blog, 2020). Critique of Honda Prize for Dr. Hinton. *Science must not allow corporate PR to distort the academic record. See also [T22].*

[HIN22] G. Hinton. The Forward-Forward Algorithm: Some Preliminary Investigations. Preprint, Google Brain, 2022. *Proposal of a biologically more plausible deep learning algorithm that—unlike backpropagation—is local in space and time. Does not mention previous related work.* [BB2][NAN1-4][NHE][MIR](Sec. 15, Sec. 17)[FWPMETA6]

[HO1] S. Hochreiter, A. S. Younger, P. R. Conwell (2001). Learning to Learn Using Gradient Descent. ICANN 2001. Lecture Notes in Computer Science, 2130, pp. 87-94.

[HO66] E. Hochstetter et al. (1966): Herrn von Leibniz' Rechnung mit Null und Eins. Berlin: Siemens AG.

[HO07] S. Hochreiter, M. Heusel, K. Obermayer. Fast model-based protein homology detection without alignment. Bioinformatics 23(14):1728-36, 2007. *Successful application of deep learning to protein folding problems, through an LSTM that was orders of magnitude faster than competing methods.*

[HRL0] J. Schmidhuber. Towards compositional learning with dynamic neural networks. Technical Report FKI-129-90, Institut für Informatik, Technische Universität München, 1990. PDF. *An RL machine gets extra command inputs of the form (start, goal). An evaluator NN learns to predict the current rewards/costs of going from start to goal. An (R)NN-based subgoal generator also sees (start, goal), and uses (copies of) the evaluator NN to learn by gradient descent a sequence of cost-minimising intermediate subgoals. The RL machine tries to use such subgoal sequences to achieve final goals. The system is learning action plans at multiple levels of abstraction and multiple time scales and solves what Y. LeCun called an "open problem" in 2022.*[LEC]

[HRL1] J. Schmidhuber. Learning to generate sub-goals for action sequences. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, Artificial Neural Networks, pages 967-972. Elsevier Science Publishers B.V., North-Holland, 1991. PDF. Extending TR FKI-129-90, TUM, 1990.

[HRL2] J. Schmidhuber and R. Wahnsiedler. Planning simple trajectories using neural subgoal generators. In J. A. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *Proc. of the 2nd International Conference on Simulation of Adaptive Behavior*, pages 196-202. MIT Press, 1992. PDF.

[HRL3] P. Dayan and G. E. Hinton. Feudal Reinforcement Learning. Advances in Neural Information Processing Systems 5, NIPS, 1992. *This work did not cite Schmidhuber's gradient-based subgoal generators for hierarchical reinforcement learning (1990).*[HRL0-2]

[HRL4] M. Wiering and J. Schmidhuber. HQ-Learning. Adaptive Behavior 6(2):219-246, 1997. PDF.

[HRLW] C. Watkins (1989). Learning from delayed rewards.

[HW1] R. K. Srivastava, K. Greff, J. Schmidhuber. Highway networks. Preprints arXiv:1505.00387 (May 2015) and arXiv:1507.06228 (July 2015). Also at NIPS 2015. *The first working very deep feedforward nets with over 100 layers (previous NNs had at most a few tens of layers). Let g, t, h, denote non-linear differentiable functions. Each non-input layer of a highway net computes g(x)x + t(x)h(x), where x is the data from the previous layer. (Like LSTM with forget gates[LSTM2] for RNNs.) Resnets[HW2] are a version of this where the gates are always open: g(x)=t(x)=const=1. Highway Nets perform roughly as well as ResNets[HW2] on ImageNet.[HW3] Variants of highway gates are also used for certain algorithmic tasks, where the simpler residual layers do not work as well.[NDR] More.*

[HW1a] R. K. Srivastava, K. Greff, J. Schmidhuber. Highway networks. Presentation at the Deep Learning Workshop, ICML'15, July 10-11, 2015. Link.

[HW2] He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. Preprint arXiv:1512.03385 (Dec 2015). *Residual nets are a version of Highway Nets[HW1] where the gates are always open: g(x)=1 (a typical highway net initialization) and t(x)=1. More.*

[HW3] K. Greff, R. K. Srivastava, J. Schmidhuber. Highway and Residual Networks learn Unrolled Iterative Estimation. Preprint arxiv:1612.07771 (2016). Also at ICLR 2017.

[HYB12] Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag., 29(6):82-97. *This work did not cite the earlier LSTM[LSTM0-6] trained by Connectionist Temporal Classification (CTC, 2006).[CTC] CTC-LSTM was successfully applied to speech in 2007[LSTM4] (also with hierarchical LSTM stacks[LSTM14]) and became the first superior end-to-end neural speech recogniser that outperformed the state of the art, dramatically improving Google's speech recognition.[GSR][GSR15][DL4] This was very different from previous hybrid methods since the late 1980s which combined NNs and traditional approaches such as hidden Markov models (HMMs).[BW][BRI][BOU] [HYB12] still used the old hybrid approach and did not compare it to CTC-LSTM. Later, however, Hinton switched to LSTM, too.[LSTM8]*

[I24] E. Ising (1925). Beitrag zur Theorie des Ferro- und Paramagnetismus. Dissertation, 1924.

[I25] E. Ising (1925). Beitrag zur Theorie des Ferromagnetismus. Z. Phys., 31 (1): 253-258, 1925. Based on [I24]. *The first non-learning recurrent NN architecture (the Ising model or Lenz-Ising model) was introduced and analyzed by physicists Ernst Ising and Wilhelm Lenz in the 1920s.[L20][I25][K41][W45][T22] It settles into an equilibrium state in response to input conditions, and is the foundation of the first well-known learning RNNs.[AMH1-2]*

[IC49] DE 833366 W. Jacobi / SIEMENS AG: Halbleiterverstärker patent filed 14 April 1949, granted 15 May 1952. *First integrated circuit with several transistors on a common substrate.*

[IC14] @CHM Blog, Computer History Museum (2014). Who Invented the IC?

[IM09] J. Deng, R. Socher, L.J. Li, K. Li, L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp.

248-255). IEEE, 2009.

[JOU17] Jouppi et al. (2017). In-Datacenter Performance Analysis of a Tensor Processing Unit. Preprint arXiv:1704.04760

[K41] H. A. Kramers and G. H. Wannier (1941). Statistics of the Two-Dimensional Ferromagnet. Phys. Rev. 60, 252 and 263, 1941.

[K56] S.C. Kleene. Representation of Events in Nerve Nets and Finite Automata. Automata Studies, Editors: C.E. Shannon and J. McCarthy, Princeton University Press, p. 3-42, Princeton, N.J., 1956.

[KAE96] L. P. Kaelbling, M. L. Littman, A. W. Moore. Reinforcement learning: a survey. Journal of AI research, 4:237-285, 1996.

[KAL59] R. Kalman. On the general theory of control systems. IRE Transactions on Automatic Control 4.3 (1959): 110-110.

[KAL59a] R. Kalman, J. Bertram. General approach to control theory based on the methods of Lyapunov. IRE Transactions on Automatic Control 4.3 (1959):20-20.

[KOE1] [21] T. Koetsier (2001). On the prehistory of programmable machines: musical automata, looms, calculators. Mechanism and Machine Theory, Elsevier, 36 (5): 589-603, 2001.

[KOH72] T. Kohonen. Correlation Matrix Memories. IEEE Transactions on Computers, C-21, p. 353-359, 1972.

[KOH82] T. Kohonen. Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43(1):59-69, 1982.

[KOH88] T. Kohonen. Self-Organization and Associative Memory. Springer, second edition, 1988.

[KOH89] H. Ritter, T. Kohonen. Self-organizing semantic maps. Biological Cybernetics, 61(4):241-254, 1989.

[KNU] D. E. Knuth, L. T. Pardo (1976). The Early Development of Programming Languages. Stanford University, Computer Science Department. PDF.

[KO2] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. Neural Networks, 10(5):857-873, 1997. PDF.

[KU] A. Küchler & C. Goller (1996). Inductive learning in symbolic domains using structure-driven recurrent neural networks. Lecture Notes in Artificial Intelligence, vol 1137. Springer, Berlin, Heidelberg.

[KUL] S. Kullback, R. A. Leibler. On information and sufficiency. The Annals of Mathematical Statistics, pages 79-86, 1951.

[L03] G. Leibniz (1703). In: Explication de l'Arithmetique Binaire / Die Mathematischen Schriften, ed. C. Gerhardt, Berlin 1879, vol.7, p.223. English link. *Leibniz documented the binary arithmetics which allow for greatly simplifiying computing hardware and are employed by virtually all modern computers. Binary number encodings per se, however, seem to date back over 4000 years.*

[L79] G. Leibniz. De Progressione dyadica Pars I. 15 March 1679. *Principles of binary computers governed by punch cards.*

[L84] G. Leibniz (1684). Nova Methodus pro Maximis et Minimis. *First publication of "modern" infinitesimal calculus.*

[L86] G. Leibniz (1686). Generales Inquisitiones de analysi notionum et veritatum. Also in Leibniz: Die philosophischen Schriften VII, 1890, pp. 236-247; translated as "A Study in the Calculus of Real Addition" (1690) by G. H. R. Parkinson, Leibniz: Logical Papers—A Selection, Oxford 1966, pp. 131-144.

[L20] W. Lenz (1920). Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern. Physikalische Zeitschrift, 21:613-615. See also [I25].

[LA14] D. R. Lande (2014). Development of the Binary Number System and the Foundations of Computer Science. The Mathematics Enthusiast, vol. 11(3):6 12, 2014. Link.

[LAN] J. L. Ba, J. R.Kiros, G. E. Hinton. Layer Normalization. arXiv:1607.06450, 2016.

[LE18] W. Lenzen. Leibniz and the Calculus Ratiocinator. Technology and Mathematics, pp 47-78, Springer, 2018.

[LEC] J. Schmidhuber (AI Blog, 2022). LeCun's 2022 paper on autonomous machine intelligence rehashes but does not cite essential work of 1990-2015. *Years ago, Schmidhuber's team published most of what Y. LeCun calls his "main original contributions:" neural nets that learn multiple time scales and levels of abstraction, generate subgoals, use intrinsic motivation to improve world models, and plan (1990); controllers that learn informative predictable representations (1997), etc. This was also discussed on Hacker News, reddit, and in the media. See tweet1. LeCun also listed the "5 best ideas 2012-2022" without mentioning that most of them are from Schmidhuber's lab, and older. See tweet2.*

[LEI07] J. M. Child (translator), G. W. Leibniz (Author). The Early Mathematical Manuscripts of Leibniz. Merchant Books, 2007. *See p. 126: the chain rule appeared in a 1676 memoir by Leibniz.*

[LEI10] O. H. Rodriguez, J. M. Lopez Fernandez (2010). A semiotic reflection on the didactics of the Chain rule. The Mathematics Enthusiast: Vol. 7 : No. 2 , Article 10. DOI: https://doi.org/10.54870/1551-3440.1191.

[LEI21] J. Schmidhuber (AI Blog, 2021). 375th birthday of Leibniz, founder of computer science.

[LEI21a] J. Schmidhuber (2021). Der erste Informatiker. Wie Gottfried Wilhelm Leibniz den Computer erdachte. (The first computer scientist. How Gottfried Wilhelm Leibniz conceived the computer.) Frankfurter Allgemeine Zeitung (FAZ), 17/5/2021. FAZ online: 19/5/2021.

[LEI21b] J. Schmidhuber (AI Blog, 2021). 375. Geburtstag des Herrn Leibniz, dem Vater der Informatik.

[LEN83] D. B. Lenat. Theory formation by heuristic search. Machine Learning, 21, 1983.

[LIL1] US Patent 1745175 by Austrian physicist Julius Edgar Lilienfeld for work done in Leipzig: "Method and apparatus for controlling electric current." First filed in Canada on 22.10.1925. *The patent describes a field-effect transistor. Today, almost all transistors are field-effect transistors.*

[LIL2] US Patent 1900018 by Austrian physicist Julius Edgar Lilienfeld: "Device for controlling electric current." Filed on 28.03.1928. *The patent describes a thin film field-effect transistor. Today, almost all transistors are field-effect transistors.*

[LIT21] M. L. Littman (2021). Collusion Rings Threaten the Integrity of Computer Science Research. Communications of the ACM, Vol. 64 No. 6, p. 43-44, June 2021.

[LSTM0] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. TR FKI-207-95, TUM, August 1995. PDF.

[LSTM1] S. Hochreiter, J. Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735-1780, 1997. PDF. Based on [LSTM0]. More.

[LSTM2] F. A. Gers, J. Schmidhuber, F. Cummins. Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10):2451-2471, 2000. PDF. *The "vanilla LSTM architecture" with forget gates that everybody is using today, e.g., in Google's Tensorflow.*

[LSTM3] A. Graves, J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18:5-6, pp. 602-610, 2005. PDF.

[LSTM4] S. Fernandez, A. Graves, J. Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. *Intl. Conf. on Artificial Neural Networks ICANN'07,* 2007. PDF.

[LSTM5] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, 2009. PDF.

[LSTM6] A. Graves, J. Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. NIPS'22, p 545-552, Vancouver, MIT Press, 2009. PDF.

[LSTM7] J. Bayer, D. Wierstra, J. Togelius, J. Schmidhuber. Evolving memory cell structures for sequence learning. Proc. ICANN-09, Cyprus, 2009. PDF.

[LSTM8] A. Graves, A. Mohamed, G. E. Hinton. Speech Recognition with Deep Recurrent Neural Networks. ICASSP 2013, Vancouver, 2013. PDF. *Based on [LSTM1-2,4,14][CTC].*

[LSTM9] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. Hinton. Grammar as a Foreign Language. Preprint arXiv:1412.7449 [cs.CL].

[LSTM10] A. Graves, D. Eck and N. Beringer, J. Schmidhuber. Biologically Plausible Speech Recognition with LSTM Neural Nets. In J. Ijspeert (Ed.), First Intl. Workshop on Biologically Inspired Approaches to Advanced Information Technology, Bio-ADIT 2004, Lausanne, Switzerland, p. 175-184, 2004. PDF.

[LSTM11] N. Beringer and A. Graves and F. Schiel and J. Schmidhuber. Classifying unprompted speech by retraining LSTM Nets. In W. Duch et al. (Eds.): Proc. Intl. Conf. on Artificial Neural Networks ICANN'05, LNCS 3696, pp. 575-581, Springer-Verlag Berlin Heidelberg, 2005.

[LSTM12] D. Wierstra, F. Gomez, J. Schmidhuber. Modeling systems with internal state using Evolino. In Proc. of the 2005 conference on genetic and evolutionary computation (GECCO), Washington, D. C., pp. 1795-1802, ACM Press, New York, NY, USA, 2005. Got a GECCO best paper award.

[LSTM13] 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. PDF.

[LSTM14] S. Fernandez, A. Graves, J. Schmidhuber. Sequence labelling in structured domains with hierarchical recurrent neural networks. In Proc. IJCAI 07, p. 774-779, Hyderabad, India, 2007 (talk). PDF.

[LSTM15] A. Graves, J. Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. *Advances in Neural Information Processing Systems 22, NIPS'22,* p 545-552, Vancouver, MIT Press, 2009. PDF.

[LSTM16] M. Stollenga, W. Byeon, M. Liwicki, J. Schmidhuber. Parallel Multi-Dimensional LSTM, With Application to Fast Biomedical Volumetric Image Segmentation. Advances in Neural Information Processing Systems (NIPS), 2015. Preprint: arxiv:1506.07452.

[LSTM17] J. A. Perez-Ortiz, F. A. Gers, D. Eck, J. Schmidhuber. Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. Neural Networks 16(2):241-250, 2003. PDF.

[LSTMPG] J. Schmidhuber (AI Blog, Dec 2020). 10-year anniversary of our journal paper on deep reinforcement learning with policy gradients for LSTM (2007-2010). *Recent famous applications: DeepMind's Starcraft player (2019) and OpenAI's dextrous robot hand & Dota player (2018)—Bill Gates called this a huge milestone in advancing AI.*

[LSTM-RL] B. Bakker, F. Linaker, J. Schmidhuber. Reinforcement Learning in Partially Observable Mobile Robot Domains Using Unsupervised Event Extraction. In Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, 2002. PDF.

[LSTMGRU] J. Chung, C. Gulcehre, K. Cho, Y. Bengio (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Preprint arXiv:1412.3555 [cs.NE]. *The so-called gated recurrent units (GRU) are actually a variant of the vanilla LSTM architecture[LSTM2] (2000) which the authors did not cite although this work[LSTM2] was the one that introduced gated recurrent units. Furthermore, Schmidhuber's team automatically evolved lots of additional LSTM variants and topologies already in 2009[LSTM7] without changing the name of the basic method. (Margin note: GRU cells lack an important gate and can neither learn to count[LSTMGRU2] nor learn simple non-regular languages;[LSTMGRU2] they also do not work as well for challenging translation tasks, according to Google Brain.[LSTMGRU3])*

[LSTMGRU2] G. Weiss, Y. Goldberg, E. Yahav. On the Practical Computational Power of Finite Precision RNNs for Language Recognition. Preprint arXiv:1805.04908.

[LSTMGRU3] D. Britz et al. (2017). Massive Exploration of Neural Machine Translation Architectures. Preprint arXiv:1703.03906

[M69] M. Minsky, S. Papert. Perceptrons (MIT Press, Cambridge, MA, 1969). *A misleading "history of deep learning" goes more or less like this: "In 1969, Minsky & Papert[M69] showed that shallow NNs without hidden layers are very limited and the field was abandoned until a new generation of neural network researchers took a fresh look at the problem in the 1980s."[S20] However, the 1969 book[M69] addressed a "problem" of Gauss & Legendre's shallow learning (circa 1800)[DL1-2] that had already been solved 4 years prior by Ivakhnenko & Lapa's popular deep learning method,[DEEP1-2][DL2] and then also by Amari's SGD for MLPs.[GD1-2] Minsky was apparently unaware of this and failed to correct it later.[HIN](Sec. I)[T22](Sec. XIII)*

[MACY51] H. Foerster, M. Mead, H. L. Teuber (eds.). Cybernetics: Circular causal and feedback mechanisms in biological and social systems. Transactions of the 7th conference. New York: J. Macy Jr. Foundation, 1951.

[MAD86] C. T. Rajagopal, M. S. Rangachari (1986). On medieval Keralese mathematics. Archive for History of Exact Sciences. 35 (2):91-99.

[MAD01] D. F. Almeida, J. K. John, A. Zadorozhnyy (2001). Keralese mathematics: Its Possible Transmission to Europe and the Consequential Educational Implications. Journal of Natural Geometry 20, 77-104, 2001.

[MAR71] D. Marr. Simple memory: a theory for archicortex. Philos Trans R Soc Lond B Biol Sci, 262:841, p 23-81, 1971.

[MAD05] Neither Newton nor Leibniz—The Pre-History of Calculus and Celestial Mechanics in Medieval Kerala. S. Rajeev, Univ. of Rochester, 2005.

[MAR15] E. Davis, G. Marcus. Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. Communications of the ACM 58.9:92-103, 2015.

[MC43] W. S. McCulloch, W. Pitts. A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, Vol. 5, p. 115-133, 1943.

[META] J. Schmidhuber (AI Blog, 2020). 1/3 century anniversary of first publication on metalearning machines that learn to learn (1987). *For its cover Schmidhuber drew a robot that bootstraps itself. 1992-: gradient descent-based neural metalearning. 1994-: Meta-Reinforcement Learning with self-modifying policies. 1997: Meta-RL plus artificial curiosity and intrinsic motivation. 2002-: asymptotically optimal metalearning for curriculum learning. 2003-: mathematically optimal Gödel Machine. 2020: new stuff!*

[META1] J. Schmidhuber. Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook. Diploma thesis, Institut für Informatik, Technische Universität München, 1987. Searchable PDF scan (created by OCRmypdf which uses LSTM). HTML. *For example, Genetic Programming (GP) is applied to itself, to recursively evolve better GP methods through Meta-Evolution. More.*

[MGC] MICCAI 2013 Grand Challenge on Mitosis Detection, organised by M. Veta, M.A. Viergever, J.P.W. Pluim, N. Stathonikos, P. J. van Diest of University Medical Center Utrecht.

[MIR] J. Schmidhuber (AI Blog, Oct 2019, updated 2021, 2022). Deep Learning: Our Miraculous Year 1990-1991. Preprint arXiv:2005.05744, 2020. *The deep learning neural networks of Schmidhuber's team have revolutionised pattern recognition and machine learning, and are now heavily used in academia and industry. In 2020-21, we celebrate that many of the basic ideas behind this revolution were published within fewer than 12 months in the "Annus Mirabilis" 1990-1991 at TU Munich.*

[MIT97] T. Mitchell. Machine Learning. McGraw Hill, 1997.

[MLP1] D. C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber. Deep Big Simple Neural Nets For Handwritten Digit Recognition. Neural Computation 22(12): 3207-3220, 2010. ArXiv Preprint. *Showed that plain backprop for deep standard NNs is sufficient to break benchmark records, without any unsupervised pre-training.*

[MLP2] J. Schmidhuber (AI Blog, Sep 2020). 10-year anniversary of supervised deep learning breakthrough (2010). No unsupervised pre-training. *By 2010, when compute was 100 times more expensive than today, both the feedforward NNs[MLP1] and the earlier recurrent NNs of Schmidhuber's team were able to beat all competing algorithms on important problems of that time. This deep learning revolution quickly spread from Europe to North America and Asia. The rest is history.*

[MM1] R. Stratonovich (1960). Conditional Markov processes. Theory of Probability and its Applications, 5(2):156-178.

[MM2] L. E. Baum, T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. The Annals of Mathematical Statistics, pages 1554-1563, 1966.

[MM3] A. P. Dempster, N. M. Laird, D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B, 39, 1977.

[MM4] J. Friedman, T. Hastie, R. Tibshirani. The elements of statistical learning, volume 1. Springer Series in Statistics, New York, 2001.

[MM5] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[MOC1] N. Metropolis, S. Ulam. The Monte Carlo method. Journal of the American Statistical Association. 44 (247): 335-341, 1949.

[MOC2] B. Abramson. The Expected-Outcome Model of Two-Player Games. Technical report, Department of Computer Science, Columbia University, 1987.

[MOC3] B. Brügmann. Monte Carlo Go. Technical report, Department of Physics, Syracuse University, 1993.

[MOC4] H. S. Chang, M. C. Fu, J. Hu, S. I. Marcus. An Adaptive Sampling Algorithm for Solving Markov Decision Processes. Operations Research. 53:126-139, 2005.

[MOC5] R. Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29â€“31, 2006.

[MOST] J. Schmidhuber (AI Blog, 2021). The most cited neural networks all build on work done in my labs. *Foundations of the most popular NNs originated in Schmidhuber's labs at TU Munich and IDSIA. (1) Long Short-Term Memory (LSTM), (2) ResNet (which is the earlier Highway Net with open gates), (3) AlexNet and VGG Net (both building on the similar earlier DanNet: the first deep convolutional NN to win image recognition competitions), (4) Generative Adversarial Networks (an instance of the much earlier Adversarial Artificial Curiosity), and (5) variants of Transformers (Transformers with linearized self-attention are formally equivalent to the much earlier Fast Weight Programmers). Most of this started with the Annus Mirabilis of 1990-1991.*[MIR]

[MOZ] M. Mozer. A Focused Backpropagation Algorithm for Temporal Pattern Recognition. Complex Systems, 1989.

[MUN87] P. W. Munro. A dual back-propagation scheme for scalar reinforcement learning. Proceedings of the Ninth Annual Conference of the Cognitive Science Society, Seattle, WA, pages 165-176, 1987.

[NAK72] K. Nakano. Associatron—A Model of Associative Memory. IEEE Transactions on Systems, Man, and Cybernetics, SMC-2:3 p. 380-388, 1972.

[NAN1] J. Schmidhuber. Networks adjusting networks. In J. Kindermann and A. Linden, editors, *Proceedings of `Distributed Adaptive Neural Information Processing', St.Augustin, 24.-25.5. 1989*, pages 197-208. Oldenbourg, 1990. Extended version: TR FKI-125-90 (revised), Institut für Informatik, TUM. PDF. *Includes the proposal of a biologically more plausible deep learning algorithm that—unlike backpropagation—is local in space and time. Based on neural nets learning to estimate gradients for other neural nets.*

[NAN2] J. Schmidhuber. Networks adjusting networks. Technical Report FKI-125-90, Institut für Informatik, Technische Universität München. Revised in November 1990. PDF.

[NAN3] Recurrent networks adjusted by adaptive critics. In *Proc. IEEE/INNS International Joint Conference on Neural Networks, Washington, D. C.*, volume 1, pages 719-722, 1990.

[NAN4] J. Schmidhuber. Additional remarks on G. Lukes' review of Schmidhuber's paper `Recurrent networks adjusted by adaptive critics'. Neural Network Reviews, 4(1):43, 1990.

[NAN5] M. Jaderberg, W. M. Czarnecki, S. Osindero, O. Vinyals, A. Graves, D. Silver, K. Kavukcuoglu. Decoupled Neural Interfaces using Synthetic Gradients. Preprint arXiv:1608.05343, 2016. *This work of DeepMind is similar to [NAN1-2].*

[NAR90] K. S. Narendra, K. Parthasarathy. Identification and control of dynamical systems using neural networks. Neural Networks, IEEE Transactions on, 1(1):4-27, 1990.

[NAS] B. Zoph, Q. V. Le. Neural Architecture Search with Reinforcement Learning. Preprint arXiv:1611.01578 (PDF), 2017. *Compare the earlier Neural Architecture Search of Bayer et al. (2009) for LSTM-like topologies.*[LSTM7]

[NASC1] J. Schmidhuber. First Pow(d)ered flight / plane truth. Correspondence, *Nature, 421 p 689, Feb 2003.*

[NASC2] J. Schmidhuber. Zooming in on aviation history. Correspondence, *Nature, vol 566, p 39, 7 Feb 2019*.

[NASC3] J. Schmidhuber. The last inventor of the telephone. Letter, *Science, 319, no. 5871, p. 1759, March 2008.*

[NASC4] J. Schmidhuber. Turing: Keep his work in perspective. Correspondence, *Nature, vol 483, p 541, March 2012*, doi:10.1038/483541b.

[NASC5] J. Schmidhuber. Turing in Context. Letter, *Science, vol 336, p 1639, June 2012*. (On Gödel, Zuse, Turing.) See also comment on response by A. Hodges (DOI:10.1126/science.336.6089.1639-a)

[NASC6] J. Schmidhuber. Colossus was the first electronic digital computer. Correspondence, *Nature, 441 p 25, May 2006.*

[NASC6a] J. Schmidhuber. Comment on "Biography: The ABC of computing" by J. Gilbey, Nature 468 p 760-761 (2010). Link.

[NASC7] J. Schmidhuber. Turing's impact. Correspondence, *Nature, 429 p 501, June 2004*

[NASC8] J. Schmidhuber. Prototype resilient, self-modeling robots. Correspondence, *Science, 316, no. 5825 p 688, May 2007.*

[NASC9] J. Schmidhuber. Comparing the legacies of Gauss, Pasteur, Darwin. Correspondence, *Nature, vol 452, p 530, April 2008.*

[NAT1] J. Schmidhuber. Citation bubble about to burst? Nature, vol. 469, p. 34, 6 January 2011. HTML.

[NDR] R. Csordas, K. Irie, J. Schmidhuber. The Neural Data Router: Adaptive Control Flow in Transformers Improves Systematic Generalization. Proc. ICLR 2022. Preprint arXiv/2110.07732.

[NEU45] J. von Neumann (1945). First Draft of a Report on the EDVAC.

[NHE] J. Schmidhuber. The Neural Heat Exchanger. Oral presentations since 1990 at various universities including TUM and the University of Colorado at Boulder. Also in In S. Amari, L. Xu, L. Chan, I. King, K. Leung, eds., Proceedings of the Intl. Conference on Neural Information Processing (1996), pages 194-197, Springer, Hongkong. Link. *Proposal of a biologically more plausible deep learning algorithm that—unlike backpropagation—is local in space and time. Inspired by the physical heat exchanger: inputs "heat up" while being transformed through many successive layers, targets enter from the other end of the deep pipeline and "cool down."*

[NGU89] D. Nguyen and B. Widrow; The truck backer-upper: An example of self learning in neural networks. In IEEE/INNS International Joint Conference on Neural Networks, Washington, D.C., volume 1, pages 357-364, 1989.

[NIL98] N. J. Nilsson. Artificial intelligence: a new synthesis. Morgan Kaufmann, 1998.

[NPMa] M. Nakamura, K. Shikano. A study of English word category prediction based on neural networks. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), p. 731-734, 1989.

[NPM] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin (2003). A Neural Probabilistic Language Model. Journal of Machine Learning Research 3, p 1137-1155, 2003. *Based on Schmidhuber & Heil's excellent 1995 neural probabilistic text model.[SNT] See also Nakamura and Shikano's 1989 word category prediction model.[NPMa]*

[NS56] A. Newell and H. Simon. The logic theory machine—A complex information processing system. IRE Transactions on Information Theory 2.3 (1956):61-79. *Compare Konrad Zuse's much earlier 1948 work on theorem proving[ZU48] based on Plankalkül, the first high-level programming language.[BAU][KNU]*

[NS59] A. Newell, J. C. Shaw, H. Simon. Report on a general problem solving program. IFIP congress, vol. 256, 1959.

[NYT1] NY Times article by J. Markoff, Nov. 27, 2016: When A.I. Matures, It May Call Jürgen Schmidhuber 'Dad'

[OAI1] G. Powell, J. Schneider, J. Tobin, W. Zaremba, A. Petron, M. Chociej, L. Weng, B. McGrew, S. Sidor, A. Ray, P. Welinder, R. Jozefowicz, M. Plappert, J. Pachocki, M. Andrychowicz, B. Baker. Learning Dexterity. OpenAI Blog, 2018.

[OAI1a] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P.

Welinder, L. Weng, W. Zaremba. Learning Dexterous In-Hand Manipulation. arxiv:1312.5602 (PDF).

[OAI2] OpenAI: C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Jozefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, S. Zhang (Dec 2019). Dota 2 with Large Scale Deep Reinforcement Learning. Preprint arxiv:1912.06680. *An LSTM composes 84% of the model's total parameter count.*

[OAI2a] J. Rodriguez. The Science Behind OpenAI Five that just Produced One of the Greatest Breakthrough in the History of AI. Towards Data Science, 2018. *An LSTM with 84% of the model's total parameter count was the core of OpenAI Five.*

[OMG0] P. T. De Chardin. The antiquity and world expansion of human culture. Man's Role in Changing the Face of the Earth. Univ. Chicago Press, Chicago, p. 103-112, 1956.

[OMG] J. Schmidhuber. An exponential acceleration pattern in the history of the most important events from a human perspective, reaching all the way back to the Big Bang. Link. Ask me Anything, Reddit/ML, 2014.

[OMG1] J. Schmidhuber (Blog, 2006). Is History Converging? Again? Based on: J. Schmidhuber. New Millennium AI and the Convergence of History. In W. Duch and J. Mandziuk, eds., Challenges to Computational Intelligence, Springer, 2006. Update of 2012 in: Singularity Hypotheses, Springer, 2013.

[OMG2] S. Berg. GRM: Brainfuck. Kiepenheuer & Witsch, 2019. *Got the Swiss Book Award 2019. Contains 2 pages on Schmidhuber's timeline of history's exponential acceleration since the Big Bang.*[OMG]

[OBJ1] K. Greff, A. Rasmus, M. Berglund, T. Hao, H. Valpola, J. Schmidhuber (2016). Tagger: Deep unsupervised perceptual grouping. NIPS 2016, pp. 4484-4492. Preprint arXiv/1606.06724.

[OBJ2] K. Greff, S. van Steenkiste, J. Schmidhuber (2017). Neural expectation maximization. NIPS 2017, pp. 6691-6701. Preprint arXiv/1708.03498.

[OBJ3] S. van Steenkiste, M. Chang, K. Greff, J. Schmidhuber (2018). Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. ICLR 2018. Preprint arXiv/1802.10353.

[OBJ4] A. Stanic, S. van Steenkiste, J. Schmidhuber (2021). Hierarchical Relational Inference. AAAI 2021. Preprint arXiv/2010.03635.

[OBJ5] A. Gopalakrishnan, S. van Steenkiste, J. Schmidhuber (2020). Unsupervised Object Keypoint Learning using Local Spatial Predictability. Preprint arXiv/2011.12930.

[OOPS2] J. Schmidhuber. Optimal Ordered Problem Solver. *Machine Learning,* 54, 211-254, 2004. PDF. HTML. HTML overview. Download OOPS source code in crystalline format.

[OUD13] P.-Y. Oudeyer, A. Baranes, 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, 2013.

[PAL80] G. Palm. On associative memory. Biological Cybernetics, 36, 1980.

[PAT17] D. Pathak, P. Agrawal, A. A. Efros, T. Darrell. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 16-17, 2017.

[PDA1] G.Z. Sun, H.H. Chen, C.L. Giles, Y.C. Lee, D. Chen. Neural Networks with External Memory Stack that Learn Context—Free Grammars from Examples. Proceedings of the 1990 Conference on Information Science and Systems, Vol.II, pp. 649-653, Princeton University, Princeton, NJ, 1990.

[PDA2] M. Mozer, S. Das. A connectionist symbol manipulator that discovers the structure of context-free languages. Proc. NIPS 1993.

[PG1] R. J. Williams. Reinforcement-learning in connectionist networks: A mathematical analysis. Technical Report 8605, Institute for Cognitive Science, University of California, San Diego, 1986

[PG3] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning 8.3-4: 229-256, 1992.

[PG2] Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999a). Policy gradient methods for reinforcement learning with function approximation. In Advances in Neural Information Processing Systems (NIPS) 12, pages 1057-1063.

[PHD] J. Schmidhuber. Dynamische neuronale Netze und das fundamentale raumzeitliche Lernproblem (Dynamic neural nets and the fundamental spatio-temporal credit assignment problem). Dissertation, Institut für Informatik, Technische Universität München, 1990. PDF. HTML.

[PLAG1] Oxford's guidance to types of plagiarism (2021). *Quote: "Plagiarism may be intentional or reckless, or unintentional."* Link.

[PLAN] J. Schmidhuber (AI Blog, 2020). 30-year anniversary of planning & reinforcement learning with recurrent world models and artificial curiosity (1990). *This work also introduced high-dimensional reward signals, deterministic policy gradients for RNNs, and* the GAN principle *(widely used today). Agents with adaptive recurrent world models even suggest a simple explanation of consciousness & self-awareness.*

[PLAN2] J. Schmidhuber. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. *Proc. IEEE/INNS International Joint Conference on Neural Networks, San Diego*, volume 2, pages 253-258, 1990. Based on TR FKI-126-90 (1990).[AC90] More.

[PLAN3] 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. San Mateo, CA: Morgan Kaufmann, 1991. PDF. Partially based on TR FKI-126-90 (1990).[AC90]

[PLAN4] J. Schmidhuber. On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models. Report arXiv:1210.0118 [cs.AI], 2015.

[PLAN5] One Big Net For Everything. Preprint arXiv:1802.08864 [cs.AI], Feb 2018.

[PLAN6] D. Ha, J. Schmidhuber. Recurrent World Models Facilitate Policy Evolution. Advances in Neural Information Processing Systems (NIPS), Montreal, 2018. (Talk.) Preprint: arXiv:1809.01999. Github: World Models.

[PM0] J. Schmidhuber. Learning factorial codes by predictability minimization. TR CU-CS-565-91, Univ. Colorado at Boulder, 1991. PDF. More.

[PM1] J. Schmidhuber. Learning factorial codes by predictability minimization. Neural Computation, 4(6):863-879, 1992. Based on [PM0], 1991. PDF. More.

[PM2] J. Schmidhuber, M. Eldracher, B. Foltin. Semilinear predictability minimization produces well-known feature detectors. Neural Computation, 8(4):773-786, 1996. PDF. More.

[PO87] J. B. Pollack. On Connectionist Models of Natural Language Processing. PhD thesis, Computer Science Department, University of Illinois, Urbana, 1987.

[PO90] J. B. Pollack. Recursive Distributed Representations. Artificial Intelligence, 46(1-2):77-105, 1990.

[POS] E. L. Post (1936). Finite Combinatory Processes—Formulation 1. Journal of Symbolic Logic. 1: 103-105. Link.

[PP] J. Schmidhuber. POWERPLAY: Training an Increasingly General Problem Solver by Continually Searching for the Simplest Still Unsolvable Problem. *Frontiers in Cognitive Science*, 2013. ArXiv preprint (2011): arXiv:1112.5309 [cs.AI]

[PPa] R. K. Srivastava, B. R. Steunebrink, M. Stollenga, J. Schmidhuber. Continually Adding Self-Invented Problems to the Repertoire: First Experiments with POWERPLAY. Proc. *IEEE Conference on Development and Learning / EpiRob 2012* (ICDL-EpiRob'12), San Diego, 2012. PDF.

[PP1] R. K. Srivastava, B. Steunebrink, J. Schmidhuber. First Experiments with PowerPlay. *Neural Networks*, 2013. ArXiv preprint (2012): arXiv:1210.8385 [cs.AI].

[PP2] V. Kompella, M. Stollenga, M. Luciw, J. Schmidhuber. Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots. Artificial Intelligence, 2015.

Relevant threads with many comments at reddit.com/r/MachineLearning, the largest machine learning forum with over 800k subscribers in 2019 (note that Jürgen Schmidhuber's name is often misspelled):

[R1] Reddit/ML, 2019. Hinton, LeCun, Bengio receive ACM Turing Award. *This announcement contains more comments about Schmidhuber than about any of the awardees.*

[R2] Reddit/ML, 2019. J. Schmidhuber really had GANs in 1990.

[R3] Reddit/ML, 2019. NeurIPS 2019 Bengio Schmidhuber Meta-Learning Fiasco. *Schmidhuber started metalearning (learning to learn—now a hot topic) in 1987[META1][META] long before Bengio who suggested in public at N(eur)IPS 2019 that he did it before Schmidhuber.*

[R4] Reddit/ML, 2019. Five major deep learning papers by G. Hinton did not cite similar earlier work by J. Schmidhuber.

[R5] Reddit/ML, 2019. The 1997 LSTM paper by Hochreiter & Schmidhuber has become the most cited deep learning research paper of the 20th century.

[R6] Reddit/ML, 2019. DanNet, the CUDA CNN of Dan Ciresan in J. Schmidhuber's team, won 4 image recognition challenges prior to AlexNet.

[R7] Reddit/ML, 2019. J. Schmidhuber on Seppo Linnainmaa, inventor of backpropagation in 1970.

[R8] Reddit/ML, 2019. J. Schmidhuber on Alexey Ivakhnenko, godfather of deep learning 1965.

[R9] Reddit/ML, 2019. We find it extremely unfair that Schmidhuber did not get the Turing award. That is why we dedicate this song to Juergen to cheer him up.

[R11] Reddit/ML, 2020. Schmidhuber: Critique of Honda Prize for Dr. Hinton

[R12] Reddit/ML, 2020. J. Schmidhuber: Critique of Turing Award for Drs. Bengio & Hinton & LeCun

[R15] Reddit/ML, 2021. J. Schmidhuber's work on fast weights from 1991 is similar to linearized variants of Transformers

[R58] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6):386. *This paper not only described single layer perceptrons, but also deeper multilayer perceptrons (MLPs). Although these MLPs did not yet have deep learning, because only the last layer learned,[DL1] Rosenblatt basically had what much later was rebranded as Extreme Learning Machines (ELMs) without proper attribution.[ELM1-2][CONN21][T22]*

[R61] Joseph, R. D. (1961). Contributions to perceptron theory. PhD thesis, Cornell Univ.

[R62] Rosenblatt, F. (1962). Principles of Neurodynamics. Spartan, New York.

[RAU1] M. Rausch. Heron von Alexandria. Die Automatentheater und die Erfindung der ersten antiken Programmierung. Diplomica Verlag GmbH, Hamburg 2012. *Perhaps the world's first programmable machine was an automatic theatre made in the 1st century by Heron of Alexandria, who apparently also had the first known working steam engine.*

[RAW] J. Schmidhuber (AI Blog, 2001). Raw Computing Power.

[RCNN] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. Preprint arXiv/1311.2524, Nov 2013.

[RCNN2] R. Girshick. Fast R-CNN. Proc. of the IEEE international conference on computer vision, p. 1440-1448, 2015.

[RCNN3] K. He, G. Gkioxari, P. Dollar, R. Girshick. Mask R-CNN. Preprint arXiv/1703.06870, 2017.

[RELU1] K. Fukushima (1969). Visual feature extraction by a multilayered network of analog threshold elements. IEEE Transactions on Systems Science and Cybernetics. 5 (4): 322-333. doi:10.1109/TSSC.1969.300225. *This work introduced rectified linear units or ReLUs, now widely used.*

[RELU2] C. v. d. Malsburg (1973). Self-Organization of Orientation Sensitive Cells in the Striate Cortex. Kybernetik, 14:85-100, 1973. *See Table 1 for rectified linear units or ReLUs. Possibly this was also the first work on applying an EM algorithm to neural nets.*

[RING91] M. B. Ring. Incremental development of complex behaviors through automatic construction of sensory-motor hierarchies. In Birnbaum, L. and Collins, G., editors, Machine Learning: Proceedings of the Eighth International Workshop, pages 343-347. Morgan Kaufmann, 1991.

[RMSP] T. Tieleman, G. E. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4.2 (2012): 26-31.

[RO98] R. Rojas (1998). How to make Zuse's Z3 a universal computer. IEEE Annals of Computing, vol. 19:3, 1998.

[ROB87] A. J. Robinson, F. Fallside. The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.

[RPG] D. Wierstra, A. Foerster, J. Peters, J. Schmidhuber (2010). Recurrent policy gradients. Logic Journal of the IGPL, 18(5), 620-634.

[RPG07] D. Wierstra, A. Foerster, J. Peters, J. Schmidhuber. Solving Deep Memory POMDPs with Recurrent Policy Gradients. *Intl. Conf. on Artificial Neural Networks ICANN'07,* 2007. PDF. *The first paper on policy gradients for LSTM. This approach has become very important in reinforcement learning.*[LSTMPG]

[RUM] DE Rumelhart, GE Hinton, RJ Williams (1985). Learning Internal Representations by Error Propagation. TR No. ICS-8506, California Univ San Diego La Jolla Inst for Cognitive Science. Later version published as: Learning representations by back-propagating errors. Nature, 323, p. 533-536 (1986). *This experimental analysis of backpropagation did not cite the origin of the method,[BP1-5] also known as the reverse mode of automatic differentiation. The paper also failed to cite the first working algorithms for deep learning of internal representations (Ivakhnenko & Lapa, 1965)[DEEP1-2][HIN] as well as Amari's work (1967-68)[GD1-2] on learning internal representations in deep nets through stochastic gradient descent. Even later surveys by the authors[DL3,3a] failed to cite the prior art.[T22]*

[RUS95] S. J. Russell, P. Norvig, J. Canny, J. M. Malik, D. D. Edwards. Artificial Intelligence: a Modern Approach, volume 2. Englewood Cliffs: Prentice Hall, 1995.

[S93] D. Sherrington (1993). Neural networks: the spin glass approach. North-Holland Mathematical Library, vol 51, 1993, p. 261-291.

[S20] T. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. PNAS, January 28, 2020. Link. *A misleading "history of deep learning" which goes more or less like this: "In 1969, Minsky & Papert[M69] showed that shallow NNs without hidden layers are very limited and the field was abandoned until a new generation of neural network researchers took a fresh look at the problem in the 1980s."[S20] However, the 1969 book[M69] addressed a "problem" of Gauss & Legendre's shallow learning (circa 1800)[DL1-2] that had already been solved 4 years prior by Ivakhnenko & Lapa's popular deep learning method,[DEEP1-2][DL2] and then also by Amari's SGD for MLPs.[GD1-2] Minsky was apparently unaware of this and failed to correct it later.[HIN](Sec. I)[T22](Sec. XIII) Deep learning research was alive and kicking in the 1960s-70s, especially outside of the Anglosphere.[DEEP1-2][GD1-3][CNN1][DL1-2][T22]*

[S80] B. Speelpenning (1980). Compiling Fast Partial Derivatives of Functions Given by Algorithms. PhD thesis, Department of Computer Science, University of Illinois, Urbana-Champaign.

[S2S] I. Sutskever, O. Vinyals, Quoc V. Le. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems (NIPS), 2014, 3104-3112.

[S59] A. L. Samuel. Some studies in machine learning using the game of checkers. IBM Journal on Research and Development, 3:210-229, 1959.

[STO51] H. Robbins, S. Monro (1951). A Stochastic Approximation Method. The Annals of Mathematical Statistics. 22(3):400, 1951.

[STO52] J. Kiefer, J. Wolfowitz (1952). Stochastic Estimation of the Maximum of a Regression Function. The Annals of Mathematical Statistics. 23(3):462, 1952.

[SA17] J. Schmidhuber. Falling Walls: The Past, Present and Future of Artificial Intelligence. Scientific American, Observations, Nov 2017.

[SCAN] J. Masci, A. Giusti, D. Ciresan, G. Fricout, J. Schmidhuber. A Fast Learning Algorithm for Image Segmentation with Max-Pooling Convolutional Networks. ICIP 2013. Preprint arXiv:1302.1690.

[SE59] O. G. Selfridge (1959). Pandemonium: a paradigm for learning. In D. V. Blake and A. M. Uttley, editors, Proc. Symposium on Mechanisation of Thought Processes, p 511-529, London, 1959.

[SHA7a] N. Sharkey (2007). A programmable robot from AD 60. New Scientist, Sept 2017.

[SHA7b] N. Sharkey (2007). A 13th Century Programmable Robot. Univ. of Sheffield, 2007. *On a programmable drum machine of 1206 by Al-Jazari.*

[SHA37] C. E. Shannon (1938). A Symbolic Analysis of Relay and Switching Circuits. Trans. AIEE. 57 (12): 713-723. Based on his thesis, MIT, 1937.

[SHA48] C. E. Shannon. A mathematical theory of communication (parts I and II). Bell System Technical Journal, XXVII:379-423, 1948. *Boltzmann's entropy formula (1877) seen from the perspective of information transmission.*

[SIN5] S. Singh, A. G. Barto, N. Chentanez. Intrinsically motivated reinforcement learning. In Advances in Neural Information Processing Systems 17 (NIPS). MIT Press, Cambridge, MA, 2005.

[SING1] V. Vinge. Marooned in Realtime. Bluejay, 1986.

[SING2] V. Vinge. The coming technological singularity: How to survive in the post-human era. Science fiction criticism: An anthology of essential writings. p. 352-363, 1993.

[SK75] D. Sherrington, S. Kirkpatrick (1975). Solvable Model of a Spin-Glass. Phys. Rev. Lett. 35, 1792, 1975.

[SKO23] T. Skolem (1923). Begründung der elementaren Arithmetik durch die rekurrierende Denkweise ohne Anwendung scheinbarer Veränderlichen mit unendlichem Ausdehnungsbereich. *Skrifter utgit av Videnskapsselskapet i Kristiania, I. Mathematisk-Naturvidenskabelig Klasse 6 (1923), 38 pp.*

[SMO86] P. Smolensky. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, pages 194-281. MIT Press, Cambridge, MA, USA, 1986.

[SMO13] L. Smolin (2013). My hero: Gottfried Wilhelm von Leibniz. The Guardian, 2013. Link. *Quote: "And this is just the one part of Leibniz's enormous legacy: the philosopher Stanley Rosen called him 'the smartest person who ever lived'."*

[SNT] J. Schmidhuber, S. Heil (1996). Sequential neural text compression. IEEE Trans. Neural Networks, 1996. PDF. *An earlier version appeared at NIPS 1995. Much later this was called a probabilistic language model.*[T22]

[SON18] T. Sonar. The History of the Priority Dispute between Newton and Leibniz. Birkhaeuser, 2018.

[ST] J. Masci, U. Meier, D. Ciresan, G. Fricout, J. Schmidhuber Steel Defect Classification with Max-Pooling Convolutional Neural Networks. Proc. IJCNN 2012. PDF. *Apparently, this was the first deep learning breakthrough in heavy industry.*

[ST61] K. Steinbuch. Die Lernmatrix. (The learning matrix.) Kybernetik, 1(1):36-45, 1961.

[ST95] W. Hilberg (1995). Karl Steinbuch, ein zu Unrecht vergessener Pionier der künstlichen neuronalen Systeme. (Karl Steinbuch, an unjustly forgotten pioneer of artificial neural systems.) Frequenz, 49(1995)1-2.

[SP16] JS interviewed by C. Stoecker: KI wird das All erobern. (AI will conquer the universe.) SPIEGEL, 6 Feb 2016. Link.

[SP93] A. Sperduti (1993). Encoding Labeled Graphs by Labeling RAAM. NIPS 1993: 1125-1132 *One of the first papers on graph neural networks.*

[SP94] A. Sperduti (1994). Labelling Recursive Auto-associative Memory. Connect. Sci. 6(4): 429-459 (1994)

[SP95] A. Sperduti (1995). Stability properties of labeling recursive auto-associative memory. IEEE Trans. Neural Networks 6(6): 1452-1460 (1995)

[SPG95] A. Sperduti, A. Starita, C. Goller (1995). Learning Distributed Representations for the Classification of Terms. IJCAI 1995: 509-517

[SPG96] A. Sperduti, D. Majidi, A. Starita (1996). Extended Cascade-Correlation for Syntactic and Structural Pattern Recognition. SSPR 1996: 90-99

[SPG97] A. Sperduti, A. Starita (1997). Supervised neural networks for the classification of structures. IEEE Trans. Neural Networks 8(3): 714-735, 1997.

[STI81] S. M. Stigler. Gauss and the Invention of Least Squares. Ann. Stat. 9(3):465-474, 1981.

[STI83] S. M. Stigler. Who Discovered Bayes' Theorem? The American Statistician. 37(4):290-296, 1983. *Bayes' theorem is actually Laplace's theorem or possibly Saunderson's theorem.*

[STI85] S. M. Stigler (1986). Inverse Probability. The History of Statistics: The Measurement of Uncertainty Before 1900. Harvard University Press, 1986.

[SV20] S. Vazire (2020). A toast to the error detectors. Let 2020 be the year in which we value those who ensure that science is self-correcting. Nature, vol 577, p 9, 2/2/2020.

[SVM1] V. N. Vapnik, A. Y. Chervonenkis. On a class of algorithms of learning pattern recognition. Automation and Telemechanics, 25.6:937, 1964.

[SVM2] C. Cortes, V. Vapnik. Support-vector networks. Machine Learning. 20 (3): 273-297, 1995

[SVM3] V. Vapnik. The nature of statistical learning theory. Springer science & business media, 1999.

[SVM4] B. Schölkopf, A. J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, 2002.

[SW1] M. Dorigo, L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1 (1): 53-66, 1997.

[SWA22] J. Swan, E. Nivel, N. Kant, J. Hedges, T. Atkinson, B. Steunebrink (2022). Work on Command: The Case for Generality. In: The Road to General Intelligence. Studies in Computational Intelligence, vol 1049. Springer, Cham. https://doi.org/10.1007/978-3-031-08020-3_6.

[T19] ACM's justification of the 2018 A.M. Turing Award (announced in 2019). WWW link. Local copy 1 (HTML only). Local copy 2 (HTML only). [T22] debunks this justification.

[T20a] J. Schmidhuber (AI Blog, 25 June 2020). Critique of 2018 Turing Award for Drs. Bengio & Hinton & LeCun. A precursor of [T22].

[T22] J. Schmidhuber (AI Blog, 2022). Scientific Integrity and the History of Deep Learning: The 2021 Turing Lecture, and the 2018 Turing Award. Technical Report IDSIA-77-21, IDSIA, Lugano, Switzerland, 2022. *Debunking [T19] and [DL3a] .*

[TD1] R. Sutton. Learning to predict by the methods of temporal differences. Machine Learning. 3 (1): 9-44, 1988.

[TD2] G. Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. Neural Computation 6.2:215-219, 1994.

[TD3] R. Sutton, A. Barto. Reinforcement learning: An introduction. Cambridge, MA, MIT Press, 1998.

[TR1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin (2017). Attention is all you need. NIPS 2017, pp. 5998-6008. *This paper introduced the name "Transformers" for a now widely used NN type. It did not cite the 1991 publication on what's now called "Transformers with linearized self-attention."[FWP0-6][TR5-6] Schmidhuber also introduced the now popular attention terminology in 1993.[ATT][FWP2][R4] See tweet of 2022 for 30-year anniversary.*

[TR2] J. Devlin, M. W. Chang, K. Lee, K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint arXiv:1810.04805.

[TR3] K. Tran, A. Bisazza, C. Monz. The Importance of Being Recurrent for Modeling Hierarchical Structure. EMNLP 2018, p 4731-4736. ArXiv preprint 1803.03585.

[TR4] M. Hahn. Theoretical Limitations of Self-Attention in Neural Sequence Models. Transactions of the Association for Computational Linguistics, Volume 8, p.156-171, 2020.

[TR5] A. Katharopoulos, A. Vyas, N. Pappas, F. Fleuret. Transformers are RNNs: Fast autoregressive Transformers with linear attention. In Proc. Int. Conf. on Machine Learning (ICML), July 2020.

[TR6] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, et al. Rethinking attention with Performers. In Int. Conf. on Learning Representations (ICLR), 2021.

[TUR] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, Series 2, 41:230-267. Received 28 May 1936. Errata appeared in Series 2, 43, pp 544-546 (1937). *2nd explicit proof that the Entscheidungsproblem (decision problem) does not have a general solution.*

[TUR1] A. M. Turing. Intelligent Machinery. Unpublished Technical Report, 1948. Link. In: Ince DC, editor. Collected works of AM Turing—Mechanical Intelligence. Elsevier Science Publishers, 1992.

[TUR21] J. Schmidhuber (AI Blog, Sep 2021). Turing Oversold. It's not Turing's fault, though.

[TUR3] G. Oppy, D. Dowe (2021). The Turing Test. Stanford Encyclopedia of Philosophy. *Quote: "it is sometimes suggested that the Turing Test is prefigured in Descartes' Discourse on the Method. (Copeland (2000:527) finds an anticipation of the test in the 1668 writings of the Cartesian de Cordemoy. Abramson (2011a) presents archival evidence that Turing was aware of Descartes' language test at the time that he wrote his 1950 paper. Gunderson (1964) provides an early instance of those who find that Turing's work is foreshadowed in the work of Descartes.)"*

[TUR3a] D. Abramson. Descartes' Influence on Turing. Studies in History and Philosophy of Science, 42:544-551, 2011.

[TUR3b] Are computers conscious?—Panpsychism with Noam Chomsky | Theories of Everything. *Mentioning the ancient "Turing Test" by Descartes.* YouTube video, 2022.

[UDRL1] J. Schmidhuber. Reinforcement Learning Upside Down: Don't Predict Rewards—Just Map Them to Actions. Preprint arXiv/1912.02875, 5 Dec 2019.

[UDRL2] R. K. Srivastava, P. Shyam, F. Mutz, W. Jaskowski, J. Schmidhuber. Training Agents using Upside-Down Reinforcement Learning. Preprint arXiv/1912.02877, 5 Dec 2019.

[UN] J. Schmidhuber (AI Blog, 2021). 30-year anniversary. 1991: First very deep learning with unsupervised or self-supervised pre-training. *Unsupervised hierarchical predictive coding (with self-supervised target generation) finds compact internal representations of sequential data to facilitate downstream deep learning. The hierarchy can be distilled into a single deep neural network (suggesting a simple model of conscious and subconscious information processing). 1993: solving problems of depth >1000.*

[UN0] J. Schmidhuber. Neural sequence chunkers. Technical Report FKI-148-91, Institut für Informatik, Technische Universität München, April 1991. PDF. *Unsupervised/self-supervised learning and predictive coding is used in a deep hierarchy of recurrent neural networks (RNNs)* *to find compact internal representations of long sequences of data, across multiple time scales and levels of abstraction. Each RNN tries to solve the pretext task of predicting its next input, sending only unexpected inputs to the next RNN above. The resulting compressed sequence representations greatly facilitate downstream supervised deep learning such as sequence classification. By 1993, the approach solved problems of depth 1000 [UN2] (requiring 1000 subsequent computational stages/layers—the more such stages, the deeper the learning). A variant collapses the hierarchy into a single deep net. It uses a so-called conscious chunker RNN which attends to unexpected events that surprise a lower-level so-called subconscious automatiser RNN. The chunker learns to understand the surprising events by predicting them. The automatiser uses a neural knowledge distillation procedure to compress and absorb the formerly conscious insights and behaviours of the chunker, thus making them subconscious. The systems of 1991 allowed for much deeper learning than previous methods. More.*

[UN1] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. Neural Computation, 4(2):234-242, 1992. Based on TR FKI-148-91, TUM, 1991.[UN0] PDF. *First working Deep Learner based on a deep RNN hierarchy (with different self-organising time scales), overcoming the vanishing gradient problem through unsupervised pre-training and predictive coding (with self-supervised target generation). Also: compressing or distilling a teacher net (the chunker) into a student net (the automatizer) that does not forget its old skills—such approaches are now widely used. More.*

[UN2] J. Schmidhuber. Habilitation thesis, TUM, 1993. PDF. *An ancient experiment on "Very Deep Learning" with credit assignment across 1200 time steps or virtual layers and unsupervised / self-supervised pre-training for a stack of recurrent NN can be found here (depth > 1000).*

[UN3] J. Schmidhuber, M. C. Mozer, and D. Prelinger. Continuous history compression. In H. Hüning, S. Neuhauser, M. Raus, and W. Ritschel, editors, *Proc. of Intl. Workshop on Neural Networks, RWTH Aachen*, pages 87-95. Augustinus, 1993.

[UN4] G. E. Hinton, R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, Vol. 313. no. 5786, pp. 504—507, 2006. PDF. *This work describes unsupervised pre-training of stacks of feedforward NNs (FNNs) called Deep Belief Networks (DBNs). It did not cite the much earlier 1991 unsupervised pre-training of stacks of more general recurrent NNs (RNNs)[UN0-3] which introduced the first NNs shown to solve very deep problems. The 2006 justification of the authors was essentially the one Schmidhuber used for the 1991 RNN stack: each higher level tries to reduce the description length (or negative log probability) of the data representation in the level below.[HIN][T22][MIR] This can greatly facilitate very deep downstream learning.[UN0-3]*

[UN5] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle. Greedy layer-wise training of deep networks. Proc. NIPS 06, pages 153-160, Dec. 2006. *The comment under reference[UN4] applies here as well.*

[UNI] Theory of Universal Learning Machines & Universal AI. Work of Marcus Hutter (in the early 2000s) on J. Schmidhuber's SNF project 20-61847: Unification of universal induction and sequential decision theory.

[URQ10] A. Urquhart. Von Neumann, Gödel and complexity theory. Bulletin of Symbolic Logic 16.4 (2010): 516-530. Link.

[VAL84] L. G. Valiant. A theory of the learnable. Communications of the ACM 27.11 (1984):1134-1142, 1984.

[VAN1] S. Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, TUM, 1991 (advisor J. Schmidhuber). PDF. More on the Fundamental Deep Learning Problem.

[VAN2] Y. Bengio, P. Simard, P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE TNN 5(2), p 157-166, 1994. *Results are essentially identical to those of Schmidhuber's diploma student Sepp Hochreiter (1991).[VAN1] Even after a common publication,[VAN3] the first author of [VAN2] published papers[VAN4] that cited only their own [VAN2] but not the original work.*

[VAN3] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, eds., A Field Guide to Dynamical Recurrent Neural Networks. IEEE press, 2001. PDF.

[VAN4] Y. Bengio. Neural net language models. Scholarpedia, 3(1):3881, 2008. Link.

[VAR13] M. Y. Vardi (2013). Who begat computing? Communications of the ACM, Vol. 56(1):5, Jan 2013. Link.

[VID1] G. Hinton. The Next Generation of Neural Networks. Youtube video [see 28:16]. GoogleTechTalk, 2007. *Quote: "Nobody in their right mind would ever suggest" to use plain backpropagation for training deep networks. However, in 2010, Schmidhuber's team in Switzerland showed[MLP1-2] that unsupervised pre-training is not necessary to train deep NNs.*

[W45] G. H. Wannier (1945). The Statistical Problem in Cooperative Phenomena. Rev. Mod. Phys. 17, 50.

[WER87] P. J. Werbos. Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. IEEE Transactions on Systems, Man, and Cybernetics, 17, 1987.

[WER89] P. J. Werbos. Backpropagation and neurocontrol: A review and prospectus. In IEEE/INNS International Joint Conference on Neural Networks, Washington, D.C., volume 1, pages 209-216, 1989.

[WI48] N. Wiener (1948). Time, communication, and the nervous system. Teleological mechanisms. Annals of the N.Y. Acad. Sci. 50 (4): 197-219. *Quote: "... the general idea of a computing machine is nothing but a mechanization of Leibniz's calculus ratiocinator."*

[WID62] Widrow, B. and Hoff, M. (1962). Associative storage and retrieval of digital information in networks of adaptive neurons. Biological Prototypes and Synthetic Systems, 1:160, 1962.

[WU] Y. Wu et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Preprint arXiv:1609.08144 (PDF), 2016. *Based on LSTM* *which it mentions at least 50 times.*

[XAV] X. Glorot, Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. Proc. 13th Intl. Conference on Artificial Intelligence and Statistics, PMLR 9:249-256, 2010.

[YB20] Y. Bengio. Notable Past Research. WWW link (retrieved 15 May 2020). Local copy (plain HTML only). *The author claims that in 1995 he "introduced the use of a hierarchy of time scales to combat the vanishing gradients issue" although Schmidhuber's publications on exactly this topic date back to 1991-93.[UN0-2][UN] The author also writes that in 1999 he "introduced, for the first time, auto-regressive neural networks for density estimation" although Schmidhuber & Heil used a very similar set-up for text compression already in 1995.[SNT]*

[ZU36] K. Zuse (1936). Verfahren zur selbsttätigen Durchführung von Rechnungen mit Hilfe von Rechenmaschinen. Patent application Z 23 139 / GMD Nr. 005/021, 1936. *First patent application describing a general, practical, program-controlled computer.*

[ZU37] K. Zuse (1937). Einführung in die allgemeine Dyadik. *Mentions the storage of program instructions in the computer's memory.*

[ZU38] K. Zuse (1938). Diary entry of 4 June 1938. *Description of computer architectures that put both program instructions and data into storage—compare the later "von Neumann" architecture [NEU45].*

[ZU48] K. Zuse (1948). Über den Plankalkül als Mittel zur Formulierung schematisch kombinativer Aufgaben. Archiv der Mathematik 1(6), 441-449 (1948). PDF. *Apparently the first practical design of an automatic theorem prover (based on Zuse's high-level programming language Plankalkül).*

[ZUS21] J. Schmidhuber (AI Blog, 2021). 80th anniversary celebrations: 1941: Konrad Zuse completes the first working general computer, based on his 1936 patent application.

[ZUS21a] J. Schmidhuber (AI Blog, 2021). 80. Jahrestag: 1941: Konrad Zuse baut ersten funktionalen Allzweckrechner, basierend auf der Patentanmeldung von 1936.

[ZUS21b] J. Schmidhuber (2021). Der Mann, der den Computer erfunden hat. (The man who invented the computer.) Weltwoche, Nr. 33.21, 19 August 2021. PDF.