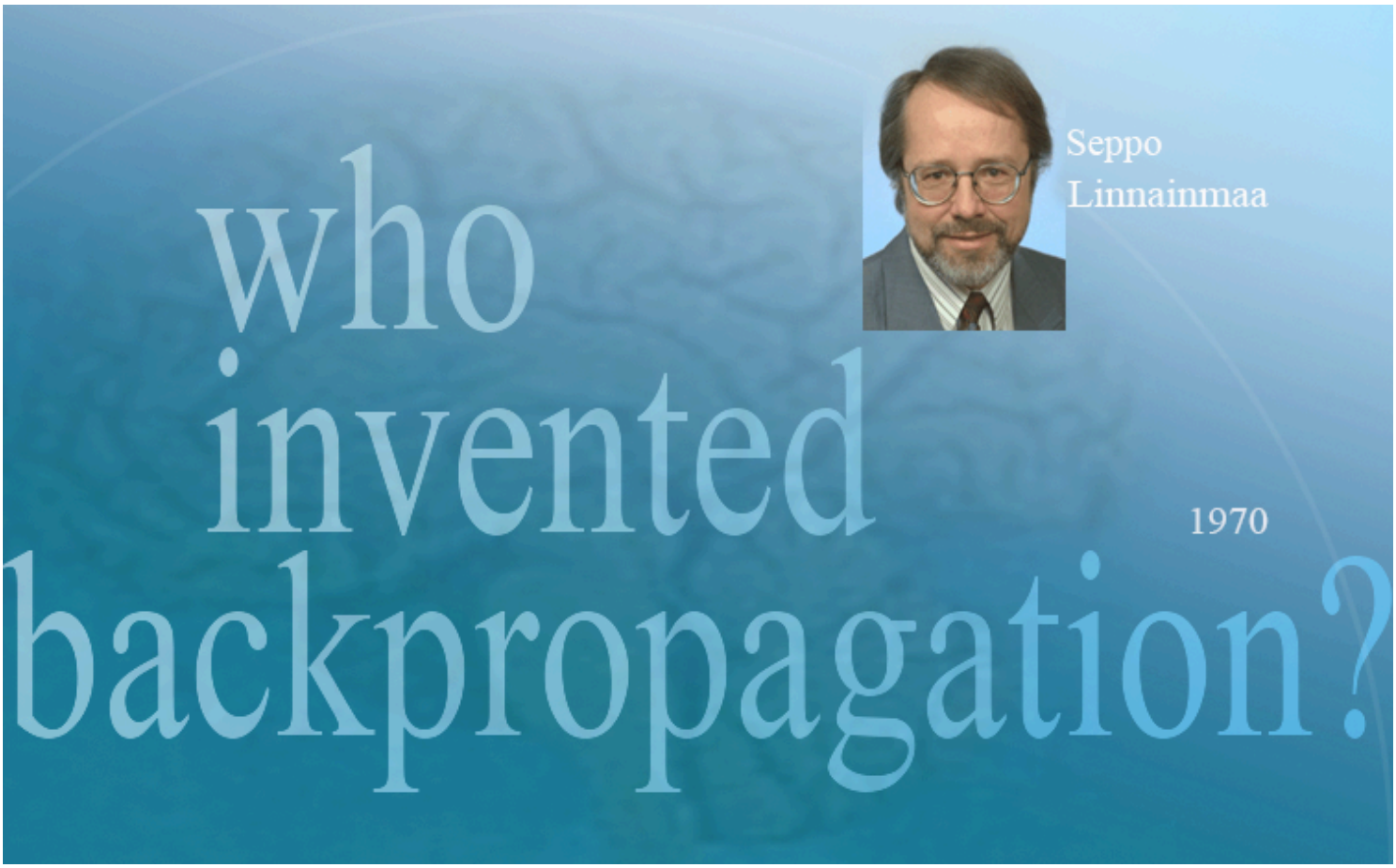

Jürgen Schmidhuber (2014, updated 2020, 2022)
Pronounce: You_again Shmidhoobuh

AI Blog
@SchmidhuberAI

Who Invented Backpropagation?

Efficient backpropagation (BP) is central to the ongoing Neural Network (NN) ReNNaissance and "Deep Learning." Who invented it?

BP's modern version (also called the reverse mode of automatic differentiation) was first published in 1970 by Finnish master student **Seppo Linnainmaa** [BP1] [R7]. **In 2020, we celebrated BP's half-century anniversary!** A precursor of BP was published by Henry J. Kelley in 1960 [BPA]—in 2020, we celebrated its 60-year anniversary.

As of 2020, it was still easy to find misleading accounts of BP's history [HIN][T22]. I had a look at the original papers from the 1960s and 70s, and talked to BP pioneers. Here is a summary based on my award-winning 2014 survey [DL1] which includes most of the references mentioned below.

The minimisation of errors through gradient descent (Cauchy 1847 [GD'], Hadamard, 1908 [GD"]) in the parameter space of complex, nonlinear, differentiable, multi-stage, NN-related systems has been discussed at least since the early 1960s, e.g., Kelley (1960) [BPA]; Bryson (1961) [BPB]; Pontryagin et al. (1961); Dreyfus (1962) [BPC]; Wilkinson

(1965); Tsypkin (1966) [GDa-b]; Amari (1967-68) [GD2,GD2a]; Bryson and Ho (1969); initially within the framework of Euler-LaGrange equations in the Calculus of Variations, e.g., Euler (1744).

Steepest descent in the weight space of such systems can be performed (Kelley, 1960 [BPA]; Bryson, 1961 [BPB]) by iterating the chain rule (Leibniz, 1676 [LEI07-10]; L'Hopital, 1696) in Dynamic Programming style (DP, e.g., Bellman, 1957 [BEL53]). A simplified derivation (Dreyfus, 1962 [BPC]) of this backpropagation method uses only the Leibniz chain rule [LEI07].

The systems of the 1960s were already efficient in the DP sense. However, they backpropagated derivative information through standard Jacobian matrix calculations from one "layer" to the previous one, without explicitly addressing either direct links across several layers or potential additional efficiency gains due to network sparsity.

Explicit, efficient error backpropagation (BP) in arbitrary, discrete, possibly sparsely connected, NN-like networks was first described in a 1970 master's thesis (Linnainmaa, 1970, 1976) [BP1][R7], albeit without reference to NNs. This kind of BP is also known as the *reverse mode of automatic differentiation* (e.g., Griewank, 2012 [BP5]), where the costs of forward activation spreading essentially equal the costs of backward derivative calculation. See early BP FORTRAN code (Linnainmaa, 1970) [BP1] and closely related but slightly later work (Ostrovskii et al., 1971). As of 2020, all modern software packages for NNs (such as Google's Tensorflow) are based on Linnainmaa's method of 1970.

BP was soon explicitly used to minimize cost functions by adapting control parameters (weights) (Dreyfus, 1973). This was followed by some preliminary, NN-specific discussion (Werbos, 1974, section 5.5.1) and a computer program for automatically deriving and implementing BP in differentiable systems (Speelpenning, 1980). The first NN-specific application of efficient BP as above was apparently described by Werbos in 1982 [BP2] (but not yet in his 1974 thesis, as is sometimes claimed).

However, already in 1967, Amari suggested to train deep multilayer perceptrons (MLPs) with many layers in non-incremental end-to-end fashion from scratch by stochastic gradient descent (SGD) [GD1], a method proposed in 1951 [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-linearily separable pattern classes. Back then compute was billions of times more expensive than today.

Compare the first deep learning MLPs called GMDH networks (Ivakhnenko and Lapa, since 1965) whose layers are incrementally grown and trained by regression analysis [DEEP1-2][R8]. These were actually the first deep NNs that learned to create hierarchical, distributed, *internal representations* of incoming data.

Additional work on backpropagation was published later (e.g., Parker, 1985; LeCun, 1985). By 1985, compute was about 1,000 times cheaper than in 1970 [BP1], and the first desktop computers became accessible in wealthier academic labs. An experimental analysis of the known method [BP1-2] by Rumelhart et al. then demonstrated that

backpropagation can yield useful internal representations in hidden layers of NNs [RUM]. At least for supervised learning, this tends to be 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].

Some ask: *"Isn't backpropagation just the chain rule of Leibniz (1676) [LEI07-10] & L'Hopital (1696)?"* No, it is the efficient way of applying the chain rule to big networks with differentiable nodes—see Sec. XII of [T22]). (There are also many inefficient ways of doing this.) It was not published until 1970 [BP1].

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], 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 showed that deep FNNs can be trained by plain backpropagation and do not at all require unsupervised pre-training for important applications [MLP1-2][MOST].

Acknowledgments

Thanks to several expert reviewers for useful comments. Since science is about self-correction, let me know under *juergen@idsia.ch* if you can spot any remaining error.

References (more in [DL1])

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

[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 his 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.

[BP6] 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.*

[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.

[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 our AI, 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.

[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). *A "survey" of*

deep learning that does not mention the pioneering works of deep learning [T22].

[DLC] J. Schmidhuber (AI Blog, June 2015). Critique of Paper by self-proclaimed "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 in [T22].*

[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 à 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.*

[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].*

[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.

[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 our 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 our "Annus Mirabilis" 1990-1991 at TU Munich.*

[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 our feedforward NNs[MLP1] and our earlier recurrent NNs 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.*

[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 my labs at TU Munich and IDSIA. Here I mention: (1) Long Short-Term Memory (LSTM), (2) ResNet (which is our earlier Highway Net with open gates), (3) AlexNet and VGG Net (both building on our similar earlier DanNet: the first deep convolutional NN to win image recognition competitions), (4) Generative Adversarial Networks (an instance of my earlier Adversarial Artificial Curiosity), and (5) variants of Transformers (Transformers with linearized self-attention are formally equivalent to my earlier Fast Weight Programmers). Most of this started with our Annus Mirabilis of 1990-1991.[MIR]*

[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.

[T20] J. Schmidhuber (June 2020). Critique of 2018 Turing Award.

[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 (v3), IDSIA, Lugano, Switzerland, 22 June 2022.

[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.

[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 failed to cite the prior art.* [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.

[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.

[UN] J. Schmidhuber (AI Blog, 2021). 30-year anniversary. 1991: First very deep learning with unsupervised pre-training. *Unsupervised hierarchical predictive coding finds compact internal representations of sequential data to facilitate downstream 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.

[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. 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 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 I 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]

[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, our team in Switzerland showed[MLP1-2] that unsupervised pre-training is not necessary to train deep NNs.*

.