SILICON VALLEY
RESEARCH
INSTITUTE
ARTIFICIAL INTELLIGENCE

Boltzmann Machines, Backpropagation, Bayesian Networks

(Copyright © 2018 Piero Scaruffi - Silicon Valley Artificial Intelligence Research Institute)

The Hopfield network proved Minsky and Papert wrong but has a problem: it tends to get trapped into what mathematicians call "local minima". Two improvements of Hopfield networks were proposed in a few months: the Boltzmann machine and backpropagation.

The Boltzmann machine was inspired by the physical process of annealing. At the same time that Hopfield introduced his recurrent neural networks, Scott Kirkpatrick at IBM introduced a stochastic method for mathematical optimization called "simulated annealing" (" Optimization by Simulated Annealing", 1983), which uses a degree of randomness to overcome local minima. This method was literally inspired by the physical process of cooling a liquid until it achieves a solid state.

In 1983 the cognitive psychologist Geoffrey Hinton, formerly a member of the PDP group at UC San Diego and now at Carnegie Mellon University, and the physicist Terry Sejnowski, a student of Hopfield but now at John Hopkins University, invented a neural network called the Boltzmann Machine that used a stochastic technique to avoid local minima, basically a Monte Carlo version of the Hopfield network ("Optimal Perceptual Inference", 1983): they used an "energy function" equivalent to Hopfield's energy function (the annealing process) but they replaced Hopfield's deterministic neurons with probabilistic neurons. Simulated annealing allows Boltzmann machines to find low energy states with high probability. Their Boltzmann Machine (with no layers) avoided local minima and converged towards a global minimum. The learning rule of a Boltzmann machine is simple, and, yet, that learning rule can discover interesting features about the training data. In reality, the Boltzmann machine is but a case of "undirected graphical models" that have long been used in statistical physics: the nodes can only have binary values (zero or one) and they are connected by symmetric connections. They are "probabilistic" because they behave according to a probability distribution, rather than a deterministic formula.

In 1986 Sejnowski trained the neural net NETtalk to pronounce English text. But there was still a major problem: the learning procedure of a Boltzmann machine is painfully slow. And it was still haunted by local minima in the case of many layers.

Helped by the same young Hinton (now at Carnegie Mellon) and by Ronald Williams, both former buddies of the PDP group, in 1986 the mathematical psychologist David Rumelhart optimized backpropagation for training multilayer (or "deep") neural networks using a "local gradient descent" algorithm that would rule for two decades, de facto a generalized delta rule ("Learning Representations by Back-propagating Errors", 1986; retitled "Learning Internal Representations by Error Propagation" as a book chapter). Error backpropagation is a very slow process and requires huge amounts of data; but backpropagation provided A.I. scientists with an efficient method to compute and adjust the "gradient" with respect to the stregths of the neural connections in a multilayer network. (Technically speaking, backpropagation is gradient descent of the mean-squared error as a function of the weights).

The world finally had a way (actually, two ways) to build multilayer neural networks to which Minsky's old critique did not apply.

Note that the idea for backpropagation came from both engineering (old cybernetic thinking about feedback) and from psychology.

At the same time, another physicist, Paul Smolensky of the University of Colorado, introduced a further optimization, the "harmonium", better known as Restricted Boltzmann Machine ("Information Processing in Dynamical Systems", 1986) because it restricts the kind of connections that are allowed between layers. The learning algorithm devised by Hinton and Sejnowski is very slow in multilayer Boltzmann machines but very fast in restricted Boltzmann machines. Multi-layered neural networks had finally become a reality. The architecture of Boltzmann machines makes it unnecessary to propagate errors, hence Boltzmann machines and all their variants do not rely on backpropagation.

These events marked a renaissance of neural networks. Rumelhart was one of the authors of the two-volume "Parallel Distributed Processing" (1986) and the International Conference on Neural Networks was held in San Diego in 1987. San Diego was an appropriate location since in 1982 Francis Crick, the British biologist who co-discovered the structure of DNA in 1953 and who now lived in southern California, had started the Helmholtz club with UC Irvine physicist Gordon Shaw (one of the earliest researchers on the neuroscience of music), Caltech neurophysiologist Vilayanur Ramachandran (later at UC San Diego), Caltech neurosurgeon Joseph Bogen (one of Roger Sperry's pupils in split-brain surgery), Caltech neurobiologists John Allman, Richard Andersen, and David Van Essen (who mapped out the visual system of the macaque monkey), Carver Mead, Terry Sejnowski and David Rumelhart. (Sad note: Rumelhart's career ended a few years later due to a neurodegenerative disease).

There were other pioneering ideas, especially in unsupervised learning.

Unsupervised learning is closely related to the problem of source separation in electrical engineering, a problem that consists in discovering the original sources of an electrical signal.

Jeanny Herault and Christian Jutten of the Grenoble Institute of Technology in France developed a method called "independent component analysis", a higher-order generalization of principal components analysis ("Space or Time Adaptive Signal Processing by Neural Network Models", 1986), later refined by Jean-Francois Cardoso at the CNRS in France ("Sources Separation Using Higher Order Moments", 1989) and by Pierre Comon at Thompson in France ("Independent Component Analysis", 1991).

In 1986 Ralph Linsker at IBM research labs in Yorktown Heights published three unsupervised models that can reproduce known properties of the neurons in the visual cortex ("From Basic Network Principles to Neural Architecture", 1986). Linsker later developed the infomax method for unsupervised learning that simplified independent component analysis ("Self-Organization in a perceptual network", 1988), which recycled one of Uttley's ideas ("Information Transmission in the Nervous System" (1979). The infomax principle is basically to maximize the mutual information that the output of a neural network processor contains about its input and viceversa.

David Zipser at UC San Diego came up with the "autoencoder", apparently from an unpublished idea by Hinton of the previous year ("Programming Networks to Compute Spatial Functions", 1986), although the term was introduced by Suzanna Becker in 1990. An autoencoder is an unsupervised neural network that is trained by backpropagation to output the input, or a very close approximation of it. In other words, it is trying to learn the identity function. In other words, an autoencoder tries to predict the input from the input. This sounds trivial, but in some cases the middle (hidden) layer ends up learning interesting facts about the data. One can also view the action of an autoencoder as a way to store an input so that it can be subsequently be retrieved as accurately as possible, i.e. the result of an autoencoder is to create a representation of the input (to "encode" the input) that allows the network to later retrieve it in a very accurate form. The autoencoder learns a representation from which the input can be reconstructed. Autoencoders are actually powerful models for capturing characteristics of data.

Dana Ballard at the University of Rochester predated deep belief networks and stacked autoencoders by 20 years when he used unsupervised learning to build representations layer by layer ("Modular Learning in Neural Networks", 1987).

Linsker’s infomax was an early application of Shannon’s information theory to unsupervised neural networks. A similar approach was tried by Mark Plumbley at Cambridge University networks ("An Information-Theoretic Approach to Unsupervised Connectionist Models", 1988).

An accelerated version of gradient descent was developed by the Russian mathematician Yurii Nesterov at the Central Economic Mathematical Institute of Moscow ("A Method of Solving a Convex Programming Problem", 1983), and it would become the most popular of the gradient-based optimization algorithms (known as "Nesterov momentum"). Later, Ning Qian at Columbia University showed similarities between Nesterov's theory and the theory of coupled and damped harmonic oscillators in physics ("On the Momentum Term in Gradient Descent Learning Algorithms", 1999).

Soon, new optimizations led to new gradient-descent methods, notably the "real-time recurrent learning" algorithm, developed simultaneously by Tony Robinson and Frank Fallside at Cambridge University ("The Utility Driven Dynamic Error Propagation Network", 1987) and Gary Kuhn at the Institute for Defense Analysis in Princeton ("A First Look at Phonetic Discrimination Using a Connectionist Network with Recurrent Links", 1987), but popularized by Ronald Williams and David Zipser at UC San Diego ("A Learning Algorithm for Continually Running Fully Recurrent Neural Networks", 1989). Paul Werbos, now at the National Science Foundation in Washington, expanded backpropagation into "backpropagation through time" ("Generalization of Backpropagation with Application to a Recurrent Gas Market Model", 1988); and variations on backpropagation through time include: the "block update" method pioneered by Ronald Williams at Northwestern University ("Complexity of Exact Gradient Computation Algorithms For Recurrent Neural Networks", 1989), the "fast-forward propagation" method by Jacob Barhen, Nikzad Toomarian and Sandeep Gulati at CalTech ("Adjoint Operator Algorithms for Faster Learning in Dynamical Neural Networks", 1991), and the "green function" method by Guo-Zheng Sun, Hsing-Hen Chen and Yee-Chun Lee at the University of Maryland ("Green's Function Method for Fast On-Line Learning Algorithm of Recurrent Neural Networks", 1992). All these algorithms were elegantly unified by Amir Atiya at CalTech and Alexander Parlos at Texas A&M University ("New Results on Recurrent Network Training", 2000).

This school of thought merged with another one that was coming from a background of statistics and neuroscience. Credit goes to Judea Pearl of UC Los Angeles for introducing Bayesian thinking into Artificial Intelligence to deal with probabilistic knowledge (“Reverend Bayes on Inference Engines", 1982). Ray Solomonoff's universal Bayesian methods for inductive inference were finally vindicated.

A kind of Bayesian network, the Hidden Markov Model, was already being used by A.I., particularly for speech recognition.

Neural networks and probabilities have something in common: neither is a form of perfect reasoning. Classical logic, based on deduction, aims to prove the truth. Neural networks and probabilities aim to approximate the truth. Neural networks are "universal approximators", as proven first by George Cybenko in 1989 at the University of Illinois ("Approximation by Superpositions of a Sigmoidal Function", 1989) and by Kurt Hornik at the Technical University in Austria, in collaboration with the economists Maxwell Stinchcombe and Halbert White of UC San Diego ("Multilayer feedforward networks are universal approximators", 1989). Cybenko and Hornik proved that neural networks can approximate any continuous function of the kind that, de facto, occurs in ordinary problems. Basically, neural networks approximate complex mathematical functions with simpler ones, which is, after all, precisely what our brain does: it simplifies the incredible complexity of the environment that surrounds us although it can only do it by approximation. Complexity is expressed mathematically by nonlinear functions. Neural networks are approximators of non-linear functions. The fact that a nonlinear function can be more efficiently represented by multilayer architectures with fewer parameters became a motivation to study multilayer architectures.

(Copyright © 2018 Piero Scaruffi - Silicon Valley Artificial Intelligence Research Institute)