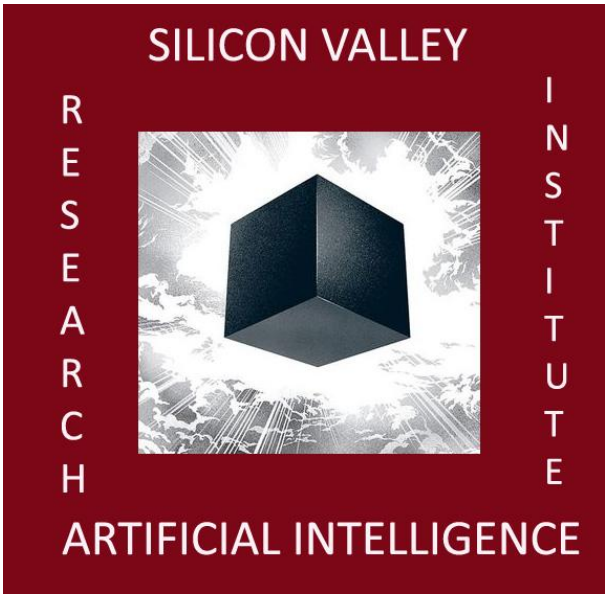

Deep Learning: a brief introduction

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

The field of neural networks did not take off until 2006, when Yee-Whye Teh at the National University of Singapore and Geoffrey Hinton at the the University of Toronto developed deep belief networks (DBNs), a fast learning algorithm for Restricted Boltzmann Machines ("A Fast Learning Algorithm for Deep Belief Nets" and "Reducing the Dimensionality of Data with Neural Networks", both 2006). Until then scientists had failed to find an effective way to train "deep" architectures (multilayer neural networks). Deep belief networks can be pre-trained in an unsupervised way, one layer at a time. Hinton's algorithms worked wonders when used on thousands of parallel processors. That's when the media started publicizing all sorts of machine-learning feats.

Deep belief networks are layered hierarchical architectures that stack Restricted Boltzmann Machines one on top of the other, each one feeding its output as input to the one immediately higher, with the two top layers forming an associative memory. The features discovered by one RBM become the training data for the next one. De facto, Hinton's deep belief networks are probabilistic models that consist of multiple layers of probabilistic reasoning. Technically speaking, Hinton's original "deep belief networks" were really "sigmoid belief networks" and only in 2009 did Hinton shift from sigmoid networks to deep Boltzmann machines.

Hinton had discovered how to create neural networks with many layers. One layer learns something and passes it on to the next one, which uses that something to learn something else and passes it on to the next layer, etc. Before 2006 deep networks performed worse than shallow networks. DBNs are still limited in one respect: they are “static classifiers”, i.e. they operate at a fixed dimensionality. However, speech or images don't come in a fixed dimensionality, but in a (wildly) variable one. They require “sequence recognition”, i.e. dynamic classifiers, that DBNs

cannot provide. One method to expand DBNs to sequential patterns is to combine deep learning with a “shallow learning architecture” like the Hidden Markov Model.

Hinton and Teh had solved the fundamental problem of multilayer networks: supervised training of deep networks is extremely difficult. Hinton and Teh figured out that it was efficient to start with unsupervised learning in which the network learns some general features. This can be repeated as many times (layers) as desired, using the output of an unsupervised learning as the input to the next one. Supervised learning takes place only on the last layer, without affecting the previous layers. Note that Hinton's work was inspired by the experiments on cats by the young Colin Blakemore at UC Berkeley. These experiments revealed that the visual cortex of the cat develops in stages, one layer at a time, as the cat is exposed to the world ("The Representation of Three-dimensional Visual Space in the Cat's Striate Cortex", 1970).

Hinton's and Teh's 2006 method for Deep belief nets came to be known as "greedy layer-wise unsupervised pre-training". Greedy algorithms are those that at every step choose the move that seems to be the most promising. Many greedy methods have been developed for classic optimization problems. Hinton and Teh had invented one to optimize the pre-training of multilayer neural networks. A few months later Yoshua Bengio at the University of Montreal introduced another method, the stacked autoencoder ("Greedy Layer-Wise Training of Deep Networks", 2007). Both techniques follow the same general procedure: they train each layer of the neural network via an unsupervised learning algorithm, taking the features produced at a level as input for the next level. The neural network learns a hierarchy of features one level at a time, and eventually it is fine-tuned in supervised fashion (backpropagation and gradient-based optimization), i.e. the last layer is a classifier.

Every neural network is, ultimately, a combination of "encoder" and "decoder": the first layers encode the input and the last layers decode it. For example, when my brain recognizes an object as an apple, it has first encoded the image into some kind of neural activity (representing shape, color, size, etc of the object), and has then decoded that neural activity as an apple.

Whereas Hinton pre-trained the network using restricted Boltzmann machines, Bengio showed that the learning algorithm of the autoencoder could also be employed as well for greedy layer-wise unsupervised pre-training. Instead of a sequence of restricted Boltzmann machines, Bengio used a sequence of autoencoders.

A stacked autoencoder is a neural network consisting of multiple layers of (sparse) autoencoders in which the outputs of each autoencoder is wired to the inputs of the successive autoencoder. Each autoencoder is designed to learn a representation of the input. Eventually, there representations are fed into a neural network in supervised learning. Therefore, just like the restricted Boltzmann machines in Hinton's deep belief nets, a stacked autoencoder learns something about the distribution of data and pre-trains the neural network that has to operate on those data. The outputs of the pre-training networks (whether stacked autoencoders or restricted Boltzmann machines) are finally passed to the real classifier (such as SVM or logistic regression). Hinton and Teh used a stochastic approach. Bengio used a deterministic approach (a restricted Boltzmann machine can be viewed as a sort of probabilistic autoencoder). More importantly, autoencoders are easier to understand, implement and tweak. The difference with

convolutional neural networks is that Hinton's and Bengio's methods work with global representations of the input, whereas LeCun's method focuses on spatially-close units (which is why convolutional neural networks are mostly used in image recognition). Autoencoders are much more general.

Alas, stacked autoencoders appeared to be not quite as accurate as deep belief networks.

Pascal Vincent of Bengio's group at Montreal later introduced the "denoising autoencoder" ("Extracting and Composing Robust Features with Denoising Autoencoders, 2008), a simple autoencoder with noise added to the training data, and he proved that such a denoising autoencoder works better at some tasks. Now stacked (de-noising) autoencoders were finally competitive with deep belief networks.

In other words: to make the reconstruction more challenging, one can use sparsity or noise. The denoising autoencoder applies noise to the input image before trying to reconstruct the clean input. The point of making the reconstruction more challenging is that the autoencoder tends to learn a more valuable representation of the data.

A third kind of step-wise pre-training of a neural network was developed by Jason Weston at NEC Labs in Princeton ("Deep Learning via Semi-supervised Embedding", 2008).

Meanwhile, Leon Bottou, now at NEC Labs in Princeton, worked on machine learning with large-scale datasets, and made "stochastic gradient descent" the optimization method of choice ("Large-Scale Machine Learning with Stochastic Gradient Descent", 2007). The stochastic gradient descent method is a drastic simplification of Rumelhart's gradient descent algorithm. The "gradient descent" method ruled until 2010 when James Martens at the University of Toronto adapted a powerful optimization method, "Hessian-free optimization" (named after the 19th century German mathematician Ludwig Hesse who invented), to neural networks ("Deep Learning via Hessian-free Optimization", 2010). Mathematicians realized that chances of success were greatly enhanced for training even very large and deep neural networks as long as they could use a lot of data. Meanwhile, in 2011 Michael Jordan's student John Duchi at UC Berkeley invented AdaGrad, a method to accelerate the learning process of the stochastic gradient descent algorithm, which would remain for a while the most popular of these "accelerators", replacing Nesterov's momentum. Its main competitors were Matt Zeiler's Adadelta ("An Adaptive Learning Rate Method", 2012), and later Adam by Diederik Kingma of the University of Amsterdam and Jimmy Ba of the University of Toronto ("A Method for Stochastic Optimization", 2015).

Later studies, for example by Andrew Ng's group at Stanford ("Rectifier Nonlinearities Improve Neural Network Acoustic Models", 2013), showed that ReLu were much important to the success of deep learning than the unsupervised layers of Hinton's and Bengio's original architectures. This generation showed that, after all, the backpropagation of the 1990s did work. It was just a matter of using large training datasets, powerful GPUs and the right kind of non-linearity.

Therefore, many scientists contributed to the “invention” of deep learning and to the resurrection of neural networks. But the fundamental contribution came from Moore's Law: between the 1980s and 2006 computers had become enormously faster, cheaper and smaller. A.I. scientists were able to implement neural networks that were hundreds of times more complex, and able to train them with millions of data. This was still unthinkable in the 1980s. Therefore what truly happened between 1986 (when Restricted Boltzmann machines were invented) and 2006 (when deep learning matured) that shifted the balance from the logical approach to the connectionist approach in A.I. was Moore's Law. Without massive improvements in the speed and cost of computers deep learning would not have happened. Deep learning owes a huge debt of gratitude to the supercharged GPUs (Graphical Processing Units) that have become affordable in the 2010s.

Credit for the rapid progress in convolutional networks goes mainly to mathematicians, who were working on techniques for matrix-matrix multiplication and made their systems available as open-source software. The software, such as UC Berkeley's Caffe, used by neural-network designers, reduces a convolution to a matrix-matrix multiplication. This is a problem of linear algebra for which seasoned mathematicians had provided solutions. Initial progress took place at the Jet Propulsion Laboratory (JPL), a research center in California operated by the California Institute for Technology (CalTech) for the space agency NASA. Charles Lawson was the head of Applied Math Group at JPL since 1965. Lawson and his employee Richard Hanson developed software for linear algebra, including software for matrix computation, that was to be applied to astronomical things like gravitational fields. In 1979, together with Fred Krogh, an expert in differential equations, they released a Fortran library called Basic Linear Algebra Subprograms (BLAS). By 1990 BLAS 3 incorporated a library for matrix-matrix operations called GEneral Matrix to Matrix Multiplication (GEMM), largely developed at Britain's Numerical Algorithms Group (instituted in 1970 as a joint project between several British universities and the Atlas Computer Laboratory). The computational "cost" of a neural network is mainly due to two kinds of layers: the layers that are fully-connected to each other and the convolutions. Both kinds entail massive multiplications of matrices; literally millions of them in the case of image recognition. Without something like GEMM no array of GPUs could perform the task.

A landmark achievement of deep-learning neural networks was published in 2012 by Alex Krizhevsky and Ilya Sutskever from Hinton's group at the University of Toronto: Krizhevsky wrote an extremely fast implementation of two-dimensional convolutions on a GPU (a network soon nicknamed AlexNet) which demonstrated that deep learning (using a convolutional neural network with five convolutional layers and Bengio's rectified linear unit) outperforms traditional techniques of computer vision after processing 200 billion images during training (1.2 million human-tagged images from ImageNet plus thousands of computer-generated variants of each). Deep convolutional neural networks became de facto standard for computer-vision systems.

At the same time Google had decided to invest into neural networks and in 2011 had funded a research team led by Stanford's professor Andrew Ng (real name Enda Wu) that came to be known the "Google Brain" group. In June 2012 Andrew Ng's group at Stanford (mainly Quoc Le and Marc'Aurelio Ranzato) demonstrated an unsupervised neural network (a large-scale autoencoder) that recognized cats in still frames of videos (not quite videos, but 10 million unlabeled images taken from videos): nobody had trained the network, i.e. nobody had told the

network that there are cats in the world and how they look like ("Building High-level Features Using Large Scale Unsupervised Learning", 2012).

Why did it take until 2012? Before the annus mirabilis of 2012 multilayer neural network had to learn too many parameters from too few labeled examples. In other words, on one hand the computers were not powerful enough and on the other hand the training datasets were not large enough. In 2012 computers became powerful enough (and Google was willing to pay for them) just when datasets had become large enough.

In 2013 Dan Ciresan of Schmidhuber's team at IDSIA achieved near-human performance in recognizing Chinese handwriting, and in 2015 Jun Sun's team at Fujitsu's Chinese laboratories surpassed human performance.

In 2013 Google hired Hinton and Facebook hired LeCun.

And in 2015 Yoshua Bengio joined Element AI, a Canadian startup founded by two veterans of the A.I. business, Nicolas Chapados and Francois Gagne, to provide customized A.I. services to businesses.

Krizhevsky's AlexNet, basically a deeper LeNet-style convolutional net, consisted of five convolutional layers followed by three fully-connected layers. It made use of Bengio's rectified linear units (ReLu) for the non-linear part, instead of a tanh or sigmoidal activation function. It was trained on Nvidia's GTX 580. In December 2013 Yann LeCun's laboratory at New York University (Pierre Sermanet, Rob Fergus and others) announced an AlexNet improvement called Overfeat.

So far deep learning had been used to classify images. The next step was to detect objects inside images. Until then the prevailing methods had been David Lowe's SIFT and Csurkas' Bag-of-Words. In 2013 Jitendra Malik's student Ross Girshick at UC Berkeley introduced Region-based Convolutional Neural Networks (or R-CNN) whose "supervised pre-training domain-specific finetuning" paradigm quickly became the new standard for object detection ("Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", 2013).

Oxford's 16-layer VGG-16, developed in 2013 by Andrew Zisserman's student Karen Simonyan at Oxford University's Visual Geometry Group (VGG), winner in 2014 of the ImageNet challenge, showed that multiple 3×3 convolutions in sequence could efficiently emulate the larger convolutions of AlexNet ("Very Deep Convolutional Networks for Large-Scale Image Recognition", 2014). The advantage was not only accuracy but also smaller GPUs.

Meanwhile, Shuicheng Yan's team at the National University of Singapore was generalizing the linear convolution operation of the convolutional network with a general nonlinear function approximator (i.e., with a neural network itself, e.g. with a multilayer perceptron) while simplifying the architecture (and reducing the number of parameters) and thus obtained the Network in Network or NiN ("Network In Network", 2014).

Later in 2014 Christian Szegedy at Google figured out a way to further optimize VGG-16 via a module inspired by the Network in Network model and called "inception module", thereby obtaining what was nicknamed the GoogLeNet or simply Inception ("Going Deeper with Convolutions", 2014). To save memory and computational costs, GoogLeNet replaced the fully-connected layer of LeCun's archetype with "global average pooling". Inception's first layer was a "depthwise separable convolution". This kind of convolutional layer, invented in 2013 by Laurent Sifre while interning at Google (and later the topic of his dissertation "Rigid-motion Scattering for Image Classification", 2014), can be trained more quickly and is less prone to the problem of "overfitting" than regular convolutions.

In 2016 Francois Chollet (of Keras fame) at Google replaced all Inception modules with depthwise separable convolutions and obtained Xception. Meanwhile, Sergey Ioffe and Christian Szegedy at Google introduced a major improvement over stochastic gradient descent called "batch normalization", a method, based on an old idea by LeCun ("Efficient Backprop", 1998), to increase the stability of a neural network and therefore accelerate the training of a deep network ("Batch Normalization", 2015).

Deep networks with many layers had an additional problem: the "degradation" problem. As one adds more layers to the network, one can incur in increasingly hostile optimization issues with the result that a simpler network might function better than a deeper one. The problem was solved in 2015 first in the Highway Network built by Schmidhuber's team at IDSIA (probably the very first neural network with over 100 layers) and then in Microsoft's 152-layer Residual Net or ResNet, designed by Jian Sun's team (including Kaiming He), a network made of so-called "residual models" to avoid issues during training (and with the same "global average pooling" of GoogLeNet). Similar to Inception, and unlike AlexNet and VGG, ResNet is a collection of building blocks. It is another "network-in-network" architecture. ResNet proved that extremely deep networks can be trained using stochastic gradient descent. Kaiming He's 2016 revision of ResNet with "identity mappings" (i.e., functions of the "f(x) = x" kind) was also easier to train ("Identity Mappings in Deep Residual Networks", 2016). To prove his point, He built a 1001-layer ResNet.

R-CNN is slow because it consists of a sequence of convolutional layers, SVMs (the object detectors) and "bounding-box regressors". Training it is expensive in both space (storage) and time, and run-time object detection is slow too. The ResNet group at Microsoft (Kaiming He, Jian Sun, etc) came up with spatial pyramid pooling networks (SPPnet), a way to optimize the computation of the convolutional layers by sharing the features that they generate, thus accelerating the run-time response of R-CNN by ten or even 100 times ("Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", 2014). They also innovated in that they accepted any image size. The other popular kinds of deep convolutional nets required a fixed input image size, i.e. cropped or warped the input images. A few months later Ross Girshick at Microsoft came up with another alternative, Fast R-CNN, three times faster in training and ten times faster at run-time ("Fast R-CNN", 2015); but not quite "real-time". Another few months and, at the end of 2015, Shaoqing Ren, working withe same crowd (He, Girshick and Sun), unveiled Faster R-CNN, a two-module network conceived precisely to achieve real-time response: a Region Proposal Network that proposes regions for a Fast R-CNN detector (it suggests "where to look"), and also shares the convolutional features in order to

optimize computation ("Towards Real-time Object Detection with Region Proposal Networks", 2015). Ali Farhadi's group at the University of Washington found another way to optimize R-CNN, using regression methods: YOLO (You Only Look Once) ("Unified, Real-time Object Detection", 2016) is not as accurate as R-CNN, but it is simpler and faster. YOLO even learns representations of objects that can be generalized.

By combining different kinds of networks, one can do more than just recognize objects. For example, Ali Farhadi's student Roozbeh Mottaghi at the Allen Institute in Seattle built the Our Forces in Scenes dataset to train two parallel convolutional neural networks (two AlexNets), one to encode the layout of a scene and the other one to capture force information, and then a recurrent neural network that takes their output and predicts the motion of the object ("What Happens If...", 2016).

Robert Jacobs' "mixture-of-experts" architecture of 1990 was refined over the years as the need for parallel architectures increased. Ronan Collobert implemented it with support vector machines ("A Parallel Mixture of SVMs for Very Large Scale Problems", 2002), Volker Tresp at Siemens in Germany used Gaussian processes ("Mixtures of Gaussian Processes", 2001), and Babak Shahbaba at UC Irvine and Radford Neal at University of Toronto modeled the "expert" as a Dirichlet process, a concept introduced by Thomas Ferguson at UCLA in 1969, which is basically a distribution over distributions ("Nonlinear Models using Dirichlet Process Mixtures", 2009). Meanwhile, Yoshua Bengio had called for "conditional computation" to speed up the work of deep neural networks: ways to reduce the number of hidden units that need to be computed in a multi-layer neural network ("Deep Learning of Representations", 2013). Jeff Dean's group at Google used mixtures of experts to achieve that goal of conditional computation and therefore dramatically increase the capacity of deep networks ("Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer", 2017).

These networks kept evolving. In 2016 Kaiming He extended ResNet to 200 layers. Szegedy produced Inception-v2 in 2015 and Inception-v3 and Inception-v4 in 2016, and in 2016 also produced a hybrid Inception-ResNet. In both Szegedy was striving for a simpler implementation without losing accuracy.

In 2016 Antonio Criminisi's team at the Microsoft labs in Britain introduced conditional networks that wed the efficiency of decision trees with the accuracy of CNNs ("Decision Forests, Convolutional Networks and the Models in-Between", 2016).

The best performing neural network of the 2012 Large Scale Visual Recognition Challenge (ILSVRC), Toronto's AlexNet, had 8 layers. The best of 2014, Oxford's VGG-16, had 19 layers. The best of 2015, Microsoft's ResNet, had 152, i.e. it was eight times deeper than the VGG network of one year earlier (ResNet was the basis for the feature of real-time translation introduced in 2015 in Microsoft's Skype product, a largely disappointing product despite the very deep structure). The trend to go deeper was propelled by two factors: 1. the hardware was getting cheaper and more powerful (Nvidia's GPUs); and 2. the mathematical theory behind deeper networks was showing the advantages. Ronen Eldan and Ohad Shamir at the Weizmann Institute of Science in Israel proved that (quote) "depth can be exponentially more valuable than width".

Deeper networks seem to be generally more accurate and require fewer connections within each layer ("The Power of Depth for Feedforward Neural Networks", 2016).

However, in 2016, while building ResNetXt, Kaiming He and Ross Girshick (both now at Facebook) realized that repetition, not depth, was the key factor in the performance of networks like VGG-16 and ResNet that are built by simply stacking modules of the same topology on top of each other. They also noted that Inception's modules shared a "split, transform and aggregate" strategy. In fact, one can reimagine the operation of a single neuron in a network as a combination of three operations: splitting, transforming, and aggregating. Imagine that the single neuron is actually a network in itself and you get a function that aggregates all the transformations of the constituent neurons. The dimension of this function is the "cardinality". Their conclusion was that the depth of a neural network is only partially interesting for improving accuracy. There is a third factor that matters, besides width and depth: the "cardinality" ("Aggregated Residual Transformations for Deep Neural Networks", 2017). Their 101-layer ResNeXt achieved better accuracy than the 200-layer ResNet while reducing the complexity to half.

Since 2012 the applications of deep learning have multiplied. Deep learning has been applied to big data, biotech, finance, health care… Countless fields hope to automate the understanding and classification of data with deep learning.

One of the most challenging applications is to recognize what is going on in a picture: there's a story hidden in those pixels, a story in which the objects are in some kind of relation for some kind of reason. We can easily look at the picture and guess what people or animals are doing or which function the objects have.

The first image-annotation system ("translating" images into sentences) was probably developed by Ryuichi Oka's team at Tsukuba Research Center in Japan ("Image-to-word Transformation Based on Dividing and Vector Quantizing Images with Words", 1999), followed by David Forsyth's team at the University of Illinois ("Object Recognition as Machine Translation", 2002). Several refinements were proposed during the 2000s, notably by Larry Davis and his student Abhinav Gupta at the University of Maryland ("Beyond Nouns", 2008).

In November 2014 simultaneously two groups (Fei-fei Li's team at Stanford and Oriol Vinyals' team at Google) announced that an A.I. system was capable of analyzing a scene and write a short description of it: "Stanford team creates computer vision algorithm that can describe photos" (Stanford's press release) and "A picture is worth a thousand (coherent) words: building a natural description of images" (posted on Google's research blog). Fei-fei Li and her Slovakian student Andrej Karpathy ("Deep Visual-Semantic Alignments for Generating Image Descriptions", 2014) had developed a mixed system of convolutional neural networks and recurrent neural networks, nicknamed Neural Talk (Karpathy was hired in 2017 by Tesla). Less publicized was the combination of recurrent neural networks and convolutional networks presented five months earlier at the International Conference on Machine Learning in Beijing by Pedro Pinheiro and Ronan Collobert from Switzerland's IDIAP ("Recurrent Convolutional Neural Networks for Scene Labeling", 2014).

Since 2012 all the main software companies have invested in A.I. startups: Apple (Siri in 2011; Boston's Perceptio and Britain-based VocalIq in 2015; San Diego's Emotient, Seattle's Turi and Indian-based Tuplejump in 2016; and Israel-based RealFace in 2017), Google (Neven, 2006; Industrial Robotics, Meka, Holomni, Bot & Dolly, DNNresearch, Schaft, Bost, DeepMind, Redwood Robotics, 2013-14; API.ai and Moodstocks, 2016; Kaggle, 2017), Amazon (San Francisco's Kiva, 2012; New York's Angel.ai, 2016; San Diego's Harvest.ai, 2017), Yahoo (LookFlow, 2013), IBM (Colorado's AlchemyAPI, 2015; plus its own Watson project), Microsoft (Britain-based Switfkey, 2016; Silicon Valley's Genee and Canada's Maluuba, 2017; plus its own Project Adam), Facebook (Israel's Face.com, 2012; Silicon Valley's Wit.ai, 2015; Belarus-based Masquerade Technologies and Switzerland-based Zurich Eye, 2017), Twitter (Boston's Whetlab, 2015; Britain-based Magic Pony, 2016), Salesforce (MetaMind, founded in 2014 in Palo Alto by Richard Socher, and Stanford's PredictionIO in 2016), Intel (Nervana and Itseex in 2016), General Electric (Berkeley's Wise.io in 2017), etc.

Numenta's cofounder Dilip George founded Vicarious in 2010 to follow a similar route to Numenta's with a slightly different model, the "recursive cortical network" (RCN), which can generalize a bit more than a deep-learning network. In 2017 Vicarious's RCN, trained on 1,406 images outperformed a deep-learning network trained on 7.9 million images.

Clarifai, founded in 2013 in New York by Rob Fergus' student Matt Zeiler of New York University who had worked on multilayer "deconvolutional network" ("Adaptive Deconvolutional Networks for Mid and High Level Feature Learning", 2011), developed the neural network that won the ImageNet competition in 2013.

Transfer learning (a term coined by Edward Thorndike a century ago) is something that we humans do naturally: we can do many things, not just one. In most cases nobody "trained" us to do what we do: we learned it by "transferring" knowledge of how to do other tasks. We can easily apply knowledge about one task to completely different tasks. Neural networks, instead, are tailored and fine-tuned for just one specific application. In 2017 a Google team (Lukasz Kaiser, Aidan Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, Jakob Uszkoreit) announced MultiModel, a neural network that had learned how to perform eight tasks but it was basically a constellation of neural networks, each specialized in a task such as image recognition, speech recognition, translation and sentence analysis. This neural network was interesting mainly because its performance on one of the eight tasks improves not only when it is trained for that task but even when it is trained for one of the other seven tasks ("One Model To Learn Them All", 2017).

As deep learning became more popular, aspiring A.I. scientists started realizing the mindboggling complexity of designing a deep-learning algorithm for a real-world application. A few labs started using A.I. techniques to choose the best A.I. technique for a given application and a new discipline, automatic machine learning, was born: Kevin Leyton-Brown's Auto-Weka (2013) at the University of British Columbia in Canada, based on Bayesian optimization; Frank Hutter's Auto-Sklearn (2015) at the University of Freiburg in Germany, based on Bayesian optimization; and Randy Olson's Tree-based Pipeline Optimization Tool or TPOT (2015) at the University of Pennsylvania based on genetic programming; and Quoc Le's and Barret Zoph's AutoML (2017) at Google, based on reinforcement learning. Meanwhile, Sergey Levine's

student Chelsea Finn at UC Berkeley worked out a general algorithm for meta-learning, i.e. for learning how to learn, "model-agnostic meta-learning" or MAML ("Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", 2017).

Several platforms for deep learning have become available as open-source software: Torch (developed by Ronan Collobert at IDIAP in Switzerland in 2002); Caffe (developed by Yangqing Jia at UC Berkeley in 2013); TensorFlow (developed by Rajat Monga's team at Google in 2015); Chainer (developed by Seiya Tokui at Preferred Networks in Japan in 2015); Microsoft Cognitive Toolkit or CNTK (2016); Keras, a deep-learning library that runs on TensorFlow, CNTK and Theano (built in 2015 by Francois Chollet at Google); and Pytorch (developed in 2016 by Soumith Chintala's team at Facebook). In 2016 OpenAI released OpenAI Gym, a toolkit for research on reinforcement learning, and in 2017 OpenAI released Roboschool, a software environment to create real-world simulations for training robots. This open-source software multiplies the number of people who can experiment with deep learning.

In 2017 the Taiwanese-based startup DT42, founded in 2015 by Bofu Chen and Tammy Yang, released on the open-source platform GitHub its BerryNet, a deep-learning system for computer vision running on a $35 Raspberry Pi, further lowering the threshold for ordinary people to develop deep-learning applications.

The real protagonist of deep learning, however, is the GPU, the graphical processing unit, originally invented to boost the speed of videogames. This humble invention constitutes the physical engine of today's multilayer neural networks, which would not be feasible without it.

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