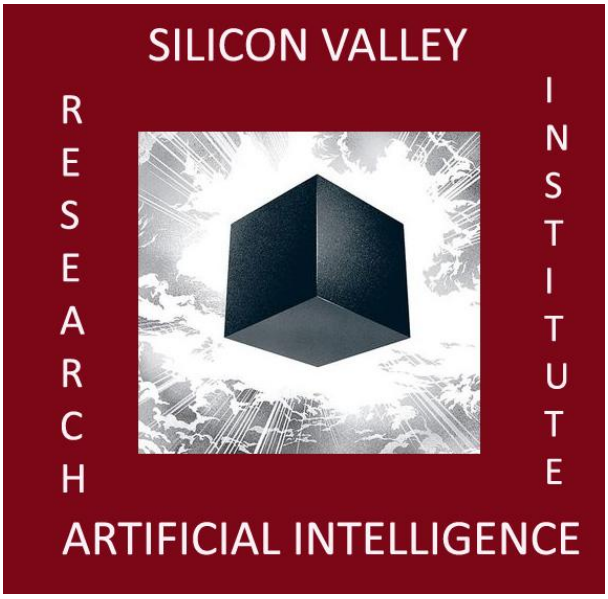

Convolutional Neural Networks: the Beginnings

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

Another thread in "deep learning" originated with convolutional networks, a kind of hierarchical multilayer networks invented in 1979 by Kunihiko Fukushima in Japan. Fukushima's Neocognitron was directly based on the 1958 studies of the cat's visual system ("Receptive Fields of Single Neurones in the Cat's Striate Cortex", 1959) by two Harvard neurobiologists, David Hubel (originally from Canada) and Torsten Wiesel (originally from Sweden). They proved that visual perception is the result of successive transformations, or, if you prefer, of propagating activation patterns: the first layer of neurons connected to the retinas detects simple features like edges, while higher layers combine these features to detect more and more complex shapes such as round objects, shapes of faces, etc. They discovered two types of neurons: simple cells, which respond to only one type of visual stimulus and behave like convolutions, and complex cells. Fukushima's system was a multi-stage architecture that mimicked those different kinds of neurons.

The same study by Hubel and Wiesel inspired "scale-invariant feature transform" (SIFT), developed by David Lowe at the University of British Columbia, that would remain the most popular algorithm in computer vision for two decades ("Object Recognition from Local Scale-Invariant Features", 1999). Lowe's original paper states: "These features share similar properties with neurons in inferior temporal cortex that are used for object recognition in primate vision."

In 1989 Alex Waibel at Carnegie Mellon University pioneered a new kind of neural network, the "time-delay" neural network ("Phoneme Recognition Using Time-Delay Neural Networks", 1989). He was working on speech recognition, i.e. on classifying phonemes, and speech signals tend to be continuous, i.e. it is not clear where a phoneme begins and ends. Time-delay neural networks introduced delays in the activation function and organized the layers around clusters, each cluster focused only on small regions of the input. His team developed one of the first

multi-lingual speech-to-speech translation systems, named Janus, in collaboration with Japan's ATR and Germany's Siemens ("A Speech-to-speech Translation System using Connectionist and Symbolic Processing Strategies", 1991).

Despite all the progress, multilayer neural nets still could not compete with traditional learning approaches such as SVMs. The first major success of neural networks came in 1989 when Yann LeCun, a former Hinton assistant at Toronto University but now at Bell Labs, applied backpropagation to convolutional networks to solve the problem of recognizing handwritten numbers ("Handwritten Digit Recognition with a Back-Propagation Network", 1989) and obtained his first convolutional neural network, later nicknamed LeNet-1, which evolved in LeNet-4 (with Leon Bottou and Yoshua Bengio) when in 1993 the National Institute of Standards and Technology released its dataset of 60,000 handwritten digits. Within a few weeks Patrice Simard (also at Bell Labs) engineered a booster version. Convolutional networks, influenced by time-delay networks, were the first success story of deep learning. Over the next few years, LeCun's team applied them to face detection ("Original Approach for the Localisation of Objects in Images," 1994) and then to reading cheques ("Gradient-based Learning Applied to Document Recognition", 1998), the latter being nicknamed LeNet-5.

By this time the architecture had stabilized in a sequence of convolutions and “pooling layers” (a more general kind of activation function), inspired by Fukushima’s Neocognitron.

Actually, the first success story in processing bank cheques were the "graph transformer networks" developed by Leon Bottou working with Bengio and LeCun ("Document Analysis with Transducers", 1996).

The seven-layer LeNet-5 represented a major improvement in computational efficiency (at a time when GPUs didn't help yet) and its architecture would remain a reference model for a decade. This architecture consisted of three components: a convolution to extract features from the image, a pooling stage to reduce the size of the representation, and a non-linearity stage in the form of either a sigmoidal activation function or a hyperbolic-tangent (or "tanh") activation function (instead of the Perceptron's step function). Mathematically speaking, the convolution is a linear operation, and some nonlinear function must be introduced to make the neural network work. Note that the feature detectors detected the presence of a feature but ignored its location inside the image. Therefore the location of a feature did not affect the classification. In other words, two eyes, a nose and a mouth would be recognized as a face even if the mouth was placed between the eyes and the nose.

The problem with LeCun's network was that Werbos-style backpropagation took almost three days to train the network for such a simple application. Clearly, this approach would not work for more complex recognition tasks.

Other neural networks were used to detect faces in 1994 by Kah-Kay Sung and Tomaso Poggio at MIT ("Example-based Learning for View-based Human Face Detection", 1994) and in 1996 by Takeo Kanade, now at Carnegie Mellon University ("Neural Network-Based Face Detection", 1996). Detecting faces is a significantly more difficult task than recognizing faces: a

face can be hidden in the middle of a very messy scene. Once you know it is a face, then it is relatively easier to find which person has the most similar face.

Until 1991, deep convolutional networks were used for recognizing isolated two-dimensional hand-written digits. Progress in recognizing three-dimensional objects had to wait until Juyang Weng's team at Michigan State University developed the Cresceptron, that used an improved technique called "max-pooling" ("Cresceptron, a Self-organizing Neural Network which Grows Adaptively", 1992).

Convolutional neural networks are not recurrent: they are feed-forward networks.

A convolutional neural network consists of several convolutional layers. Each convolution layer consists of a convolution or filtering stage (the “simple cell”), a detection stage, and a pooling stage (the “complex cell”), and the result of each convolutional layer is in the form of “feature maps”, and that is the input to the next convolutional layer. The last layer is a classification module.

The detection stage of each convolutional layer is the middleman between simple cells and complex cells and provides the nonlinearity of the traditional multi-layer neural network. Traditionally, this nonlinearity was provided by a mathematical function called "sigmoidal", but in 2011 Yoshua Bengio ("Deep Sparse Rectifier Networks") introduced a more efficient function, the “rectified linear unit”, also inspired by the brain, that have the further advantage of avoiding the "gradient vanishing" problem of sigmoidal units.

Every layer of a convolutional network detects a set of features, starting with large features and moving on to smaller and smaller features. Imagine a group of friends subjected by you to a simple game. You show a picture to one of them, and allow him to provide a short description of the picture to only another one and using only a very vague vocabulary; for example: an object with four limbs and two colors. This new person can then summarize that description in a more precise vocabulary to the next person; for example a four-legged animal with black and white stripes. Each person is allowed to use a more and more specific vocabulary to the next person. Eventually, the last person can only utter names of objects, and hopefully correctly identifies the picture because, by the time it reaches this last person, the description has become fairly clear (e.g. the mammal whose skin is black and white, i.e. the zebra).

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