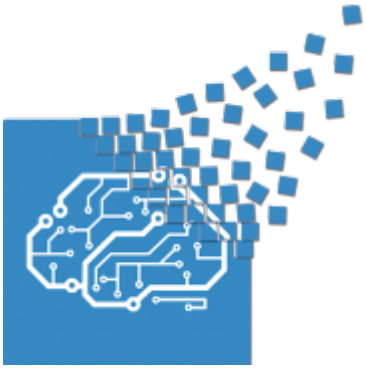
What if...

By Tomaso Poggio June 26, 2015



The background: DCLNs (Deep Convolutional Learning Networks) are doing very well

Over the last 3 years and increasingly so in the last few months, I have seen supervised DCLNs — feedforward and recurrent — do more and more of everything quite well. They seem to learn good representations for a growing number of speech and text problems (for a review by the pioneers in the field see LeCun, Bengio, Hinton, 2015). More interestingly, it is increasingly clear, as I will discuss later, that instead of being trained on millions of labeled examples they can be trained in *implicitly supervised* ways. This breakthrough in machine learning triggers a few dreams. *What if we have now the basic answer to how to develop brain-like intelligence and its basic building blocks?*

Why it may be true

There are several reasons to have been skeptical of neural networks old claims. But I think that I see now possible answers to all of them. I list the corresponding questions here, together with answers (which are, in part, conjectures) ranked in terms of increasing (personal) interest.

- **What is the tradeoff of nature vs. nurture (for neural networks)?** I think that a version of the Baldwin (1896) effect (rediscovered by Hinton and Nowlan, 1987) provides a good framework for an answer. If an organism inherits the machinery that can learn a task (important for survival) from examples provided by the environment, then evolution only needs to discover the machinery and compile it into the genes. It does not need to discover and compile the full, detailed solution to the task — and never will, because of the lack of sufficient evolutionary pressure. This argument suggests that evolution determines the architecture of the network, for instance it may determine slightly different deep learning architectures for different sensory tasks with respect to number of layers, connectivity and pooling parameters — say in visual cortex vs. auditory cortex. Thus the tradeoff between nature and nurture, which applies

in general, could mean for deep learning networks that weights are learned through visual experience, both unsupervised and implicitly supervised (see later), whereas the architecture and its parameters are determined by the genes. In other words, a *provocative conjecture* is that evolution may have done what neural networks architects are doing these days: choose the problem, and thereby the training data set, tinker with the architecture — number of layers, which ones are convolutional, how much to downsample at each layer, etc. — until a good one is found for the problem at hand (say vision vs. speech). What is left for learning during the life of an organism is to set a large number of weights, which may require millions of — explicitly or implicitly — supervised examples. Of course, there are closely related, more likely scenarios, of which a good example are recent versions of Ullman's *visual routines*. In this scenario a few basic tasks learned early on by appropriate networks are used as building blocks for more complex abilities.

- **How powerful are multilayer feedforward architectures such as DCLNs?** There are several answers, both old and new, but also some important open questions. Consider multilayer networks in which each layer performs the following operation on the vector input from the earlier layer:

$$x^i = \sum c_j \sigma(w^j \circ x^{i-1})$$

From the computer science point of view, feedforward multilayer networks are equivalent to finite state machines running for a finite number of time steps (see Shalev- Schwartz, 2014 for a recent account and Poggio and Reichardt, 1980 for an old one). This result holds for almost any fixed nonlinearity in each layer: it holds when the nonlinearity is polynomial, for instance quadratic, in which case multiple layers may be needed for each time step. Feedforward networks are equivalent to cascades without loops (with a finite number of stages) and all other forms of loop free cascades (i.e. McCulloch-Pitts nets without loops, perceptrons, analog perceptrons, linear threshold machines). Finite state machines, cascades with loops, and difference equation systems, which are Turing equivalent, are, thus more powerful than multilayer architectures with a finite number of layers. The latter networks, however, are practically universal computers, since every machine we can build can be approximated as closely as we like by defining sufficiently many stages or a sufficiently complex single stage (think about the polynomial example). Recurrent networks as differential equations are of course Turing universal.

For a brief overview, which includes also analog perceptrons and multilayer polynomial machines see Poggio and Reichardt (1980).

From the function approximation point of view, the choice of the appropriate representation for a given computation and thus the associated complexity - in terms of number of layers and connectivity (measured by *order* for perceptrons, *p-order* for polynomial networks) — depends on the type of elementary components and connections that are available. This perspective on the complexity of networks is related to Hilbert's 13th problem, which concerns the possibility of representing functions of several variables as superposition of functions of a smaller number of variables. In fact, in 1957 Kolmogorov (1963) and Arnold (1963) have shown that exactly 2-hidden layers networks can always represent any continuous function of n variables. Thus the Kolmogorov result shows that the number of layers cannot be taken as a full measure of complexity. In fact, it is well-known that:

- one-hidden layer networks with appropriate, “universal” nonlinearities can represent arbitrarily well any continuous function, possibly using a very large number of units; the units in the network have in general order (in the perceptron sense) infinity, that is their receptive field includes all the inputs (the whole “retina”)
- with nonlinearities which are linear and quadratic, any continuous function can be approximated arbitrarily well, but the number of layers must be arbitrarily large

In general, both number of layers and order — that is the maximum number of inputs for each unit among all units in a layer — play a role in measuring the complexity of a network, depending on the type of nonlinear operations. For instance, DCLNs have finite, small order in the first layer(s), whereas fully connected DLNs have typically order infinity (in all layers).

Clearly the class of multilayer networks with specified nonlinearity is no less powerful than the class of one-hidden layer networks. With a *universal* nonlinearity one-hidden layer networks can approximate any nonlinear mapping (in fact they can approximate all functions of d variables defined at points, therefore functions learnable with labeled examples[1]). A prototypical machine that bridges between the computer science and the function

[1] Therefore with a certain degree of smoothness, because of Sobolev lemma.

approximation point of view is the set of *polynomial perceptrons* mentioned above.

The open questions concern a precise characterization of the conditions under which multilayer networks with universal nonlinearities are "better" than one-hidden layer networks. Our current conjecture is that, if the *d*-dimensional space of inputs has a "compositional" structure in which low-dimensional subspaces can be well represented by a small number of prototypical parts, multilayer networks can achieve much better compression (in terms of total number of bits required to encode the ensemble of weights). Details of the argument can be found in Anselmi et al. (2015). A rigorous solution of this question would be a major accomplishment for deep learning.

- **What guarantees that SGD converges to good solutions?** SGD is not guaranteed to converge to a good solution and it often does not. However large amounts of data often lead to impressive solutions that generalize, that is predict, well. This has been the greatest surprise for me: the dimensionality of the space over which optimization takes place is huge. A large, but not exponentially large, number of data seems sufficient to lead in many cases to "predictive" solutions.

- **Are DCLN consistent with the metaphor of the brain as an interpolating look-up table?** I always thought that a zero-order metaphor for thinking about the sensory parts of the brain and how it may have evolved from basic synaptic plasticity was the "*look-up table*" metaphor. In this context the sensory brain could be characterized in terms of memory-based computation[2]. This is described in an old paper "How the brain might work" and related there to radial basis function networks and to the ability to generalize from a set of examples (Poggio, 1990). It is not completely obvious that DCLNs can be consistent with view but this is indeed the conclusion of Anselmi et al. (2015). Under the assumption of normalized inputs, each layer in a DCLN can be equivalent with appropriate weights to sets of Gaussian-like radial units. The center of each unit represents a memory — either one example (input component of the input- output example pair) or a prototype representing a cluster of examples in effectively a hierarchical memory (see later).

- **Is the lack of a theory a problem for DCLNs?** It is deeply unsatisfactory to have a potential explanation for the brain and to be unable to understand it.

[2] This is less of a constraint that it may sound: the basic logical operations of our computers can be described in terms of look-up tables.

However, this is clearly not a good reason for rejecting the engineering use of neural networks, especially after demonstrations of their good performance in many settings. Of course, theories are desirable for many reasons, including the need for a guide to future progress and improvements. I am also more optimistic that a satisfactory theoretical framework is developing along the lines sketched earlier. The invariance properties of convolutional networks can be characterized mathematically and extended beyond the translation group. The selectivity properties for each layer also have now mathematical foundations (Anselmi et al., 2015 and references there)[3]. The missing part for a full theory is formal proofs for the role of hierarchies in architectures such as HMAX and DCLNs (see conjecture mentioned above).

- **Is supervised training with millions of labeled examples biologically plausible?** As I mentioned earlier, setting weights requires millions of supervised examples — at least if one looks at the case of Imagenet[4]. I claimed for some time now that training with millions of labeled examples was biologically implausible ("*not the way children learn to distinguish a dog from a cat*"). While strictly speaking I was correct, I think now that there is a relatively simple solution. Labels are of course arbitrary. What is clearly important for the organism is to be able to group together (with the implicit label "same identity") images of the same object (or for classification, of the same object class). I call this *implicit labeling*: explicit labels are not provided but there is contextual or other information that allows implicit labeling. Several plausible ways are available to biological organisms, especially during development, to implicitly label in this ways images and other sensory stimuli such as sounds and words. For images, time continuity is a powerful and primitive cue for implicit labeling. In fact time continuity was proposed by i-theory to learn invariance to transformations during development (Poggio, 2011) by associating together images of different transformations of the same template. The same strategy is used by Wang and Gupta (CVPR, 2015) by tracking patches of images in

[3] The original i-theory deals with networks such as HMAX in which there is only unsupervised learning of templates for invariance. The theory also describes convolutional layers in DCLNs. Its recent extension (Anselmi et al., 2015) deals with nonlinearities such as linear rectifier units. It can be applied to convolutional and non-convolutional layers in DCLNs trained with backpropagation or other supervised techniques.

[4] The original Imagenet networks could benefit from additional shift and scale invariances (and other invariances), which are relatively easy to learn in a implicitly supervised way according to i-theory and related empirical work (see also HMAX).

videos. Many other strategies[5] also are used: an example among many is egomotion information replacing labels in videos (Malik et al., 2015). Simple bootstrapping schemes are also effective under rather general conditions. An early example, among many others, is provided by Poggio, Fahle and Edelman (1992; see also Fahle, M., S. Edelman, and T. Poggio. *Vision Research*, 1995) for perceptual learning (in particular vernier acuity). Perceptual learning takes place even in the absence of feedback to the subject, that is without labels (Poggio *et al*, 1992; see also Weiss *et al*, 1993) showed that networks requiring supervised examples could still account for the data if used in a bootstrapping mode, in which very few initial examples correctly labeled could be sufficient to classify novel examples that are sufficiently similar to them. They used HyperBF models (equivalent to neural networks for normalized inputs) in which learning takes place in two distinct ways: *unsupervised* learning is required to establish, create or tune the "centers" whereas *supervised* learning determines the appropriate “synaptic" weights for the coefficients. The first type of learning does not require labels while the second does. Given this, here are some back of the envelope calculations of how many unlabeled images a baby could get during the first year of life. Suppose one saccade per second per 8 hours per 360 days: this gives in the order of ~10M images. Even if only a small fraction, like 10%, could be implicitly labeled this would provide a sufficiently rich training set as suggested by the empirical results on Imagenet. It is interesting to speculate about developments paths for a sensory modality such as vision of an organism such as human baby. In a first stage of life, templates (corresponding to DCLN weights) could be random images; invariances could be learned in this stage according to i-theory by simply storing images of the same object during shift and scale transformations. Such a simple approach provides invariance and sufficient selectivity, though suboptimal, as shown by theory (references in Anselmi et al., 2015) and by empirical studies by Mutch and LeCun. In a second stage, implicitly supervised learning may take over, interspersed with occasional fully supervised learning[6]. It must be said that several of us do not yet believe that single DLN architectures can deal with challenging cognitive tasks. However, it may be possible to build a mind with architectures that use as modules DCL networks. The question is then whether

[5] In cases of tasks accomplished over time, reinforcement learning can also be described as weakly supervised (as suggested by G. Roig).

[6] Only some special form of data augmentation, which was previously called *virtual examples* (see Niyogi, Girosi and Poggio, 1998) can be plausibly used by biological organisms.

and how the architecture and the single modules may be learned from experience and, if yes, how.

What else is left to be done?

Much of the answers above are just based on plausibility. They are meant to show how the main problems that I had could be solved. Thus they require verification and modifications. In addition, they suggest quite a bit of further work. Here is a sample of work to be done.

- The field of ISL (implicitly supervised learning) seems important from the biological perspective — and eventually for engineering as well. It will probably to be more empirical in character than theoretical. It may include different forms of bootstrapping. It may yield implications for evolutionary theory via the Baldwin effect.

- As I mentioned, the most surprising empirical finding related to neural networks and backpropagation is the effectiveness of stochastic gradient descent for large networks and large data. Theoretical insights on the underlying mathematical reasons would be highly desirable.

- On the neuroscience side the question of the learning mechanisms is even more interesting. Since backpropagation does not seem biologically plausible, it is critical to find alternatives to it that are neurally plausible. As pointed out in several papers over the last 3 decades, a number of learning schemes may replace backpropagation while being usually more inefficient (see for instance the reinforcement-like algorithms described by Seung, 2003). Needless to say, a proposal in this direction that receives experimental support would be a major step in understanding the brain. There are non- trivial questions. For instance, do invariant (e.g. convolutional) layers require an explicit weight-sharing among synapses or does this happen automatically as described by i-theory? If there is weight sharing what could the biophysical mechanism be? The difference amounts to whether invariance is coded by the genes and has been learned by evolution or is learned during development. I conjecture the existence of a specific plasticity mechanism as follows: *the single cell model of the Hubel and Wiesel module suggested in i-theory and shown in Figure 1 could naturally support the type of plasticity required by weight-sharing*. The hypothetical mechanism for plasticity in cortical pyramidal cells consists of

two parts:

- a Foldyak-type Hebbian-like mechanism establishes (and possibly maintains) the wiring from LGN inputs to simple-cell-like subunits on the complex cell.
- synapses in all subunits of the complex cell receive the same facilitation or depression signal for plastic changes.

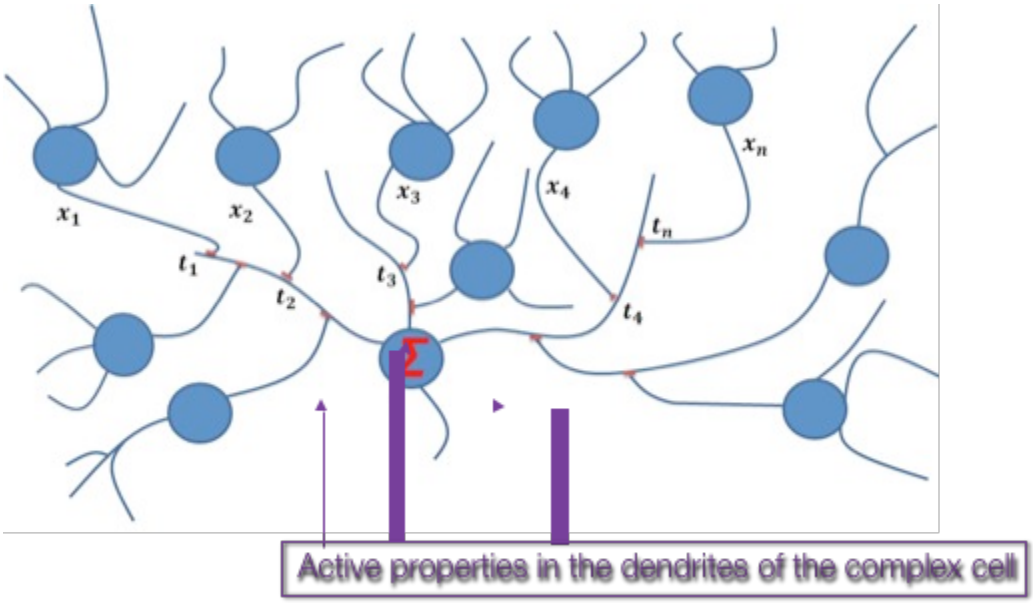


Figure 1. *A model of a complex cell in which separate dendritic subunits play the role of simple cells in i-theory. If the synapses in these subunits represent the same template in nearby positions in the visual field similar changes in all of them correspond to weight- sharing in a DCLN.*

- The network of face tuned cortical patches may represent the best opportunity for solving a problem — face identification and other aspects of face perception — that is likely to be a prototype for the problem of object recognition.

- The idea of an *autoencoder* (see for instance Hinton and Salakhutdinov, 2006) as an extension of Principal Component Analysis remains a theoretically interesting area in itself and as a way to decrease the number of labeled examples. It may also have neural implementations in terms of Hebbian mechanisms. It is related to separately trained analysis and synthesis networks for video compression and computer graphics (Beymer and Poggio, 1996).

- Because of the success of deep learning we know that Deep Convolutional Learning Networks like HMAX can be trained effectively with large numbers of labeled examples. This may be biologically plausible if we can show that ILEs could be used to the same effect. What needs to be done is to train, with a plausible number of ILEs, biologically plausible multilayer architectures similar and better than HMAX (Serre et al, 2007). For instance, for visual cortex it would be important to take into account known parameters, such as receptive field sizes, related range of pooling for known invariance and especially eccentricity dependence of RFs — all for each of the known cortical areas in the ventral stream. A comparison of behavioral performance and cell tuning properties to human and primate data for several different visual tasks would represent a major *Turing*++ test for this class of models.

- In Poggio and Smale (2003) we wrote "*A comparison with real brains offers another, and probably related, challenge to learning theory. The "learning algorithms" we have described in this paper correspond to one-layer architectures. Are hierarchical architectures with more layers justifiable in terms of learning theory? It seems that the learning theory of the type we have outlined does not offer any general argument in favor of hierarchical learning machines for regression or classification. This is somewhat of a puzzle since the organization of cortex -- for instance visual cortex -- is strongly hierarchical. At the same time, hierarchical learning systems show superior performance in several engineering applications...*". Twelve years later, a most interesting theoretical question that still remains open, both for machine learning and neuroscience, is indeed *why hierarchies*.

Acknowledgment

The author is supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF – 1231216. Part of the writing was done in Singapore at the Institute for Infocomm supported by the REVIVE project and on a beach in Bali. Thanks for constructive comments to Lorenzo Rosasco, Fabio Anselmi, Shimon Ullman, Cheston Tan, Leyla Isik, Gemma Roig and Xavier Boix.

A few relevant references

Agrawal, P., Carreira, J., Malik, J. (2015). Learning to See by Moving, CVPR

Anselmi, F., Rosasco, L., Tan, C., and Poggio, T., "Deep Convolutional Networks are Hierarchical Kernel Machines", (2015) CBMM Memo No. 035

Arnold, V.I. (1963), Representation of continuous functions of three variables by the superposition of continuous functions of two variables. Am. Math. Soc. Trans., Ser. 2, 28

Baldwin, J.M. (1896), A new factor in evolution. American Naturalist, 30, 441-451.

Beymer, D. and T. Poggio (1996). Image Representation for Visual Learning, Science, 272, 1905-1909.

Fahle, M., S. Edelman, and T. Poggio (1995). Fast Perceptual Learning in Visual Hyperacuity, Vision Research, Vol. 35, 21, 3003-3013.

Hinton, G.E., and Nowlan, S.J. (1987), How learning can guide evolution. Complex Systems, 1, 495-502.

Hinton, G. & Salakhutdinov, R. (2006) Reducing the Dimensionality of Data with Neural Networks, Science, 28 July

Kolmogorov, A.N. (1963) On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition. Am. Math. Soc. TransI. 28,55-59

LeCun, Y., Bengio, Y., Hinton, G. (2015), Deep learning. Nature 521, 436–444.

Niyogi, P., T. Poggio, and F. Girosi, (1998). Incorporating Prior Information in Machine Learning by Creating Virtual Examples. In: IEEE Proceedings on Intelligent Signal Processing, Vol. 86, No 11, 2196-2209,

Poggio, T. (1990), A Theory of How the Brain Might Work, In: Proceedings of Cold Spring Harbor Symposia on Quantitative Biology, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 4, 899-910.

Poggio, T. (2011), sections with J. Mutch, J.Z. Leibo and L. Rosasco, The Computational Magic of the Ventral Stream: Towards a Theory, Nature Precedings, doi:10.1038/npre.2011.6117.1

Poggio, T., M. Fahle and S. Edelman, (1992). Fast Perceptual Learning in Visual Hyperacuity, Science, 256, 1018-1021.

Poggio and W. Reichardt (1980). On the Representation of Multi-Input Systems: Computational Properties of Polynomial Algorithms. Biological Kybernetik, 37, 167-186.

Poggio, T. and S. Smale, (2003) The Mathematics of Learning: Dealing with Data, Notices of the American Mathematical Society (AMS), Vol. 50, No. 5, 537-544.

Serre, T., G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich and T. Poggio (2007), A Quantitative Theory of Immediate Visual Recognition. In: Progress in Brain Research

Seung, S. (2003). Learning in Spiking Neural Networks by Reinforcement of Stochastic Synaptic Transmission, Neuron, Vol. 40, 1063–1073

Shalev-Shwartz, Shai and Ben-David, S. (2014) Understanding Machine Learning: From Theory to Algorithms, Cambridge eBooks

Wang and Gupta (2015), Unsupervised Learning of Visual Representations using Videos, CVPR