The Science and the Engineering of Intelligence

Tomaso Poggio

Center for Biological and Computational Learning
McGovern Institute for Brain Research at MIT
Department of Brain & Cognitive Sciences
CSAIL
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
Engineering of Intelligence: recent successes
Intelligence: engineering
Recent progress in AI
Demis Hassabis, master of the new machine age
Murad Ahmed

The creator of the AI game-playing program makes all the right moves, writes Murad Ahmed
Why now: very recent progress in AI
Mobileye
20 years ago: MIT and Daimler
CBMM: motivations

Key recent advances in the engineering of intelligence have their roots in basic science of the brain.
The same hierarchical architectures in the cortex, in models of vision and in Deep Learning networks

Desimone & Ungerleider 1989; vanEssen+Movshon
The race for Intelligence

- The science of intelligence was at the roots of today’s engineering success

- …we need to make another basic effort on it
  - for the sake of basic science
  - for the engineering of tomorrow
Mission: We aim to make progress in understanding intelligence — that is in understanding how the brain makes the mind, how the brain works and how to build intelligent machines.

CBMM’s main goal is to make progress in the science of intelligence which enables better engineering of intelligence.
Interdisciplinary

Cognitive Science

Machine Learning
Computer Science

Neuroscience
Computational Neuroscience

Science + Technology of Intelligence
Centerness: collaborations across different disciplines and labs

MIT
Boyden, Desimone, Kaelbling, Kanwisher, Katz, Poggio, Sassanfar, Saxe, Schulz, Tenenbaum, Ullman, Wilson, Rosasco, Winston

Harvard
Blum, Kreiman, Mahadevan, Nakayama, Sompolinsky, Spelke, Valiant

Rockefeller
Freiwald

Allen Institute
Koch

UCLA
Yuille

Stanford
Goodman

Cornell
Hirsh

Hunter
Epstein, Sakas, Chodorow

Wellesley
Hildreth, Conway, Wiest

Puerto Rico
Bykhovaskaia, Ordonez, Arce Nazario

Howard
Manaye, Chouikha, Rwebargira
Recent Stats and Activities

- IIT Metta, A*star Tan, Hebrew U. Shashua, MPI Buelthoff
- Genoa U. Verri, Weizmann Ullman
- City U. HK Smale

Recent Stats and Activities

- Google, IBM, Microsoft, Siemens, Schlumberger, GE
- MobilEye, Honda, Boston Dynamics, Orcam, Nvidia, Rethink Robotics

Third CBMM Summer School, 2016
EAC members

Pietro Perona, Caltech
Charles Isbell, Jr., Georgia Tech
Joel Oppenheim, NYU

Lore McGovern, MIBR, MIT
David Siegel, Two Sigma

Demis Hassabis*, DeepMind
Marc Raibert, Boston Dynamics

Kobi Richter, Medinol
Judith Richter, Medinol
Dan Rockmore, Dartmouth
Susan Whitehead, MIT Corporation
Fei-Fei Li, Stanford
CBMM

Brains, Minds and Machines Summer School at Woods Hole: our flagship initiative
Brains, Minds and Machines Summer School

In 2016: 302 applications for 35 slots
Brains, Minds and Machines Summer School

Broad introduction to research on human and machine intelligence

- computation, neuroscience, cognition
- research methods and current results
- lecture videos on CBMM website
- summer 2015 course materials to be published on MIT OpenCourseWare

List of speakers*:

Tomaso Poggio  Gabriel Kreiman  Nancy Kanwisher  Boris Katz
Winrich Freiwald  Matthew Wilson  Josh Tenenbaum  L Mahadevan
Elizabeth Spelke  Rebecca Saxe  Shimon Ullman  Laura Schulz
Ken Nakayama  Patrick Winston  Lorenzo Rosasco  Ethan Meyers
Amnon Shashua  James DiCarlo  Larry Abbott  Aude Oliva
Dorin Comaniciu  Tom Mitchell  Eero Simoncelli  Eddy Chang
Demis Hassabis  Josh McDermott

* CBMM faculty, industrial partners
Learning by Doing: Lab Work & Joint Student Projects
An example project across thrusts: face recognition
A project across thrusts: face recognition

Winrich Freiwald and Doris Tsao
A project across thrusts: face recognition
A project across thrusts: face recognition
Another project

When and why are deep networks better than shallow networks?

Work with Hrushikeshl Mhaskar; initial parts with L. Rosasco and F. Anselmi
Computation in a neural net

\[ f(x) = f_L(\ldots f_2(f_1(x))) \]
<table>
<thead>
<tr>
<th>mite</th>
<th>container ship</th>
<th>motor scooter</th>
<th>leopard</th>
</tr>
</thead>
<tbody>
<tr>
<td>mite</td>
<td>container ship</td>
<td>motor scooter</td>
<td>leopard</td>
</tr>
<tr>
<td>black widow</td>
<td>lifeboat</td>
<td>go-kart</td>
<td>jaguar</td>
</tr>
<tr>
<td>cockroach</td>
<td>amphibian</td>
<td>moped</td>
<td>cheetah</td>
</tr>
<tr>
<td>tick</td>
<td>fireboat</td>
<td>bumper car</td>
<td>snow leopard</td>
</tr>
<tr>
<td>starfish</td>
<td>drilling platform</td>
<td>golfcart</td>
<td>Egyptian cat</td>
</tr>
<tr>
<td>grille</td>
<td>mushroom</td>
<td>cherry</td>
<td>Madagascar cat</td>
</tr>
<tr>
<td>convertible</td>
<td>agaric</td>
<td>dalmatian</td>
<td>squirrel monkey</td>
</tr>
<tr>
<td>grille</td>
<td>mushroom</td>
<td>grape</td>
<td>spider monkey</td>
</tr>
<tr>
<td>pickup</td>
<td>jelly fungus</td>
<td>elderberry</td>
<td>titi</td>
</tr>
<tr>
<td>beach wagon</td>
<td>gill fungus</td>
<td>currant</td>
<td>indri</td>
</tr>
<tr>
<td>fire engine</td>
<td>dead-man’s-fingers</td>
<td>currant</td>
<td>howler monkey</td>
</tr>
</tbody>
</table>
Computation in a neural net

Rectified linear unit (ReLU)

\[ g(y) = \max(0, y) \]
Gradient descent

$$\arg\min_w \sum_i \ell(z_i, f(x_i; w)) = L(w)$$

One iteration of gradient descent:

$$w^{t+1} = w^t - \eta_t \frac{\partial L(w^t)}{\partial w}$$

learning rate
Stochastic gradient descent

$L(w)$
What can you do with them?

Parse images

Long et al., CVPR 2015

Model perception

Vig et al., CVPR 2014

Model the brain

Yamins et al., PNAS 2014

Beat humans at Atari games

Mnih et al., Nature 2015
Hierarchical feedforward models of the ventral stream do "work"
Convolutional networks

“Hubel-Wiesel” models include

Hubel & Wiesel, 1959: Fukushima, 1980, Wallis & Rolls, 1997; Mel, 1997; LeCun et al 1998; Riesenhuber & Poggio, 1999; Thorpe, 2002; Ullman et al., 2002; Wersing and Koerner, 2003; Serre et al., 2007; Freeman and Simoncelli, 2011....
Hierarchical **feedforward** models of the ventral stream do “work”
The same hierarchical architectures in the cortex, in the models of vision and in Deep Learning networks.
Using goal-driven deep learning models to understand sensory cortex

Daniel L K Yamins¹,² & James J DiCarlo¹,²
Using goal-driven deep learning models to understand sensory cortex

Daniel L K Yamins¹,² & James J DiCarlo¹,²

(a) IT single-site neural predictivity ( % explained variance) vs. Categorization performance (balanced accuracy).

(b) HCNN top hidden layer response prediction vs. IT neural response.

(c) Category neural predictivity (% explained variance) for Monkey V4 (n = 128) and Monkey IT (n = 166).

(d) Human IT (IMRI) vs. HCNN model.

(e) R2C visual correlation (Kendall's τ) for Human V1-V3 vs. Human IT.
The same hierarchical architectures in the cortex, in the models of vision and in Deep Learning networks.
DLNNs: two main scientific questions

When and why are deep networks better than shallow networks?

Why does SGD work so well for deep networks? Could unsupervised learning work as well?

Work with Hrushikeshl Mhaskar; initial parts with L. Rosasco and F. Anselmi
How do the learning machines described by classical learning theory -- such as kernel machines -- compare with brains?

One of the most obvious differences is the ability of people and animals to learn from very few examples ("poverty of stimulus" problem).

A comparison with real brains offers another, related, challenge to learning theory. Classical "learning algorithms" correspond to one-layer architectures. The cortex suggests a hierarchical architecture.

Thus...are hierarchical architectures with more layers the answer to the sample complexity issue?
Deep and shallow networks: universality

Theorem Shallow, one-hidden layer networks with a nonlinear $\phi(x)$ which is not a polynomial are universal. Arbitrarily deep networks with a nonlinear $\phi(x)$ (including polynomials) are universal.

$$g(x) = \sum_{i=1}^{r} c_i |\langle w_i, x \rangle + b_i|_+$$
Classical learning theory and Kernel Machines
(Regularization in RKHS)

\[
\min_{f \in H} \left[ \frac{1}{\ell} \sum_{i=1}^{\ell} V(f(x_i) - y_i) + \lambda \| f \|_K^2 \right]
\]

implies

\[
f(x) = \sum_i \alpha_i K(x, x_i)
\]

Equation includes splines, Radial Basis Functions and Support Vector Machines (depending on choice of V).


For a review, see Poggio and Smale, *The Mathematics of Learning*, Notices of the AMS, 2003
Classical kernel machines are equivalent to shallow networks.

Kernel machines...

\[ f(x) = \sum^l_i c_i K(x, x_i) + b \]

can be “written” as shallow networks: the value of K corresponds to the “activity” of the “unit” for the input and the correspond to “weights”
Deep and shallow networks: universality

**Theorem** Shallow, one-hidden layer networks with a nonlinear $\phi(x)$ which is not a polynomial are universal. Arbitrarily deep networks with a nonlinear $\phi(x)$ (including polynomials) are universal.

$$g(x) = \sum_{i=1}^{r} c_i \langle w_i, x \rangle + b_i$$
Deep and shallow networks

- Thus depth is not needed to for approximation

\[ g(x) = \sum_{i=1}^{r} c_i \langle w_i, x \rangle + b_i \]
Deep and shallow networks

- Thus depth is not needed for approximation.
- Conjecture: depth may be more effective for certain classes of functions.

\[ g(x) = \sum_{i=1}^{r} c_i \langle w_i, x \rangle + b_i \]
When is deep better than shallow

\[ f(x_1, x_2, \ldots, x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8))) \]
Theorem:
why and when are deep networks better than shallow network?

\[ f(x_1, x_2, \ldots, x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8))) \]
Theorem:
why and when are deep networks better than shallow network?

\[ f(x_1, x_2, ..., x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8))) \]

\[ g(x) = \sum_{i=1}^{r} c_i |\langle w_i, x \rangle + b| \]
Theorem:
why and when are deep networks better than shallow network?

\[ f(x_1, x_2, ..., x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8))) \]

**Theorem (informal statement)**
Suppose that a function of \( d \) variables is compositional. Both shallow and deep network can approximate \( f \) equally well. The number of parameters of the shallow network depends exponentially on \( d \) as \( O(\varepsilon^{-d}) \) with the dimension whereas for the deep network depends linearly on \( d \) that is \( O(d\varepsilon^{-2}) \)
Shallow vs deep networks

Theorem 1. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, and not a polynomial on any subinterval of $\mathbb{R}$.

(a) For $f \in W^{NN}_{r,d}$

$$\text{dist}(f, \mathcal{S}_n) = \mathcal{O}(n^{-r/d}).$$

(b) For $f \in W^{NN}_{H,r,d}$

$$\text{dist}(f, \mathcal{D}_n) = \mathcal{O}(n^{-r/2}).$$

This is the best possible estimate (n-width result)
Similar results for VC dimension of shallow vs deep networks

Theorem 5. The VC-dimension of the shallow network with $N$ units is bounded by $(d + 2)N^2$; VC-dimension of the binary tree network with $n(d - 1)$ units is bounded by $4n^2(d - 1)^2$. 

Poggio, Anselmi, Rosasco, 2015
Theorem

Suppose that a function of $d$ variables is compositional. Both shallow and deep network can approximate $f$ equally well. The number of parameters of the shallow network depends exponentially on $d$ as $O(\varepsilon^{-d})$ with the dimension whereas for the deep network depends linearly on $d$ that is $O(d\varepsilon^{-2})$.

New Proof. Linear combinations of 6 units provides an indicator function; $k$ partitions for each coordinates require $6kn$ units in one layer. The next layer computes the entries in the 2D table corresponding to $g(x_1, x_2)$; they also correspond to tensor products. Two layers with $6kn + (6kn)^2$ units represent one of the $g$ functions. For convolutional nets total units is $(l (6kn + (6kn)^2))$.
Our theorem implies directly other known results

- A classical theorem [Hastad, 1987] shows that deep circuits are more efficient in representing certain Boolean functions than shallow circuits. Hastad proved that highly-variable functions (in the sense of having high frequencies in their Fourier spectrum) in particular the parity function cannot even be decently approximated by small constant depth circuits.

- The main result of [Telgarsky, 2016, Colt] says that there are functions with many oscillations that cannot be represented by shallow networks with linear complexity but can be represented with low complexity by deep networks.
Corollary

Our main theorem implies Hastad and Telgarsky theorems.

Use our theorem with Boolean variables. Consider the parity function
\[ x_1 x_2 \ldots x_d \]
which is compositional. Q.E.D

For the second part, consider for instance the real-valued polynomial
\[ x_1 x_2 \ldots x_d \]
defined on the cube \((-1, 1)^d\). This is a compositional functions that changes signs a lot. Q.E.D.
The curse of dimensionality, the blessing of compositionality

The previous examples show three different kinds of “sparsity” that allow a blessed representation by deep networks with a much smaller number of parameters than by shallow networks. This state of affairs motivates the following general definition of relative dimension. Let $d_n(W)$ be the non-linear $n$-width of a function class $W$. For the unit ball $\mathcal{B}_{\gamma, q}$ of the class $\mathcal{W}_{\gamma, q}$ as defined in Section 3.2, the Bernstein inequality proved in [17] leads to $d_n(\mathcal{B}_{\gamma, q}) \sim n^{-\gamma/(2q)}$. In contrast, for the unit ball $\mathcal{GB}_\gamma$ of the class we have shown that $d_n(\mathcal{GB}_\gamma) \leq cn^{-\gamma/(2d)}$, where $d = \max_{v \in V} d(v)$.

Generalizing, let $V$, $W$ be compact subsets of a metric space $X$, and $d_n(V)$ (respectively, $d_n(W)$) be their $n$–widths. We define the relative dimension of $d_n(V)$ with respect to $d_n(W)$ by

$$D(V, W) = \limsup_{n \to \infty} \frac{\log d_n(V)}{\log d_n(W)}. \quad (6.3)$$

Thus, $D(\mathcal{GB}_\gamma, \mathcal{B}_{\gamma, q}) \leq d/q$. This leads us to say that $V$ is parsimonious with respect to $W$ if $D(V, W) \ll 1$. 
The curse of dimensionality, 
the blessing of compositionality

For compositional functions deep networks — but not shallow ones — can avoid the curse of dimensionality, that is the exponential dependence on the dimension of the network complexity and of its sample complexity.
Why are compositional functions important?

They seem to occur in computations on text, speech, images...why?

Conjecture (with Max Tegmark)

The hamiltonians of physics induce compositionality in natural signals such as images
Remarks

1. A binary tree net is a good proxy for ResNets

2. Scalable algorithms and compositional functions

4. Invariance and pooling

6. Sparse functions and Boolean functions
Convolutional Deep Networks (no pooling like in ResNets)

Similar theorems apply to the network on the left and the network on the right in terms of # parameters.
Hyper deep residual networks: a binary tree net is a good mathematical proxy
Remarks

1. A binary tree net is a good proxy for ResNets

2. Scalable algorithms and compositional functions

4. Invariance and pooling

6. Sparse functions and Boolean functions
Definition 1. For integer $m \geq 2$, an operator $H_{2m}$ is shift-invariant if $H_{2m} = H'_m \oplus H''_m$ where $\mathbb{R}^{2m} = \mathbb{R}^m \oplus \mathbb{R}^m$, $H' = H''$ and $H' : \mathbb{R}^m \mapsto \mathbb{R}^{m-1}$. An operator $K_{2M} : \mathbb{R}^{2M} \rightarrow \mathbb{R}$ is called scalable and shift invariant if $K_{2M} = H_2 \circ \cdots H_{2M}$ where each $H_{2k}$, $1 \leq k \leq M$, is shift invariant.

We observe that scalable shift-invariant operators $K : \mathbb{R}^{2m} \mapsto \mathbb{R}$ have the structure $K = H_2 \circ H_4 \circ H_6 \cdots \circ H_{2m}$, with $H_4 = H'_2 \oplus H'_2$, $H_6 = H''_2 \oplus H''_2 \oplus H''_2$, etc. Thus the structure of a shift-invariant, scalable operator consists of several layers; each layer consists of identical blocks;
Qualitative arguments for compositional functions in vision

- Images require algorithms of the compositional function type
- Recognition in clutter requires computations with compositional functions
Remarks

1. A binary tree net is a good proxy for ResNets

2. Scalable algorithms and compositional functions

4. Invariance and pooling: interpretation of nodes in binary tree

6. Sparse functions and Boolean functions
Comment on i-theory

- i-theory is not essential for today theorem; it represents a further analysis of convolutional networks and extensions of them.

- i-theory characterizes how convolution and pooling in multilayer networks reduces sample complexity (→ Lorenzo).

- Theorems about extending invariance beyond position invariance and how to learn it from the environment (→ Lorenzo).
Remarks

1. A binary tree net is a good proxy for ResNets
2. Scalable algorithms and compositional functions
4. Invariance and pooling
6. Sparse functions and Boolean functions
Sparse functions

$$Q(x, y) = (Ax^2y^2 + Bx^2y + Cxy^2 + Dx^2 + 2Exy + Fy^2 + 2Gx + 2Hy + I)^{10}.$$
More remarks

• Functions that are not compositional/sparse may not be learnable by deep networks

• Deep, non-convolutional, densely connected networks are not better than shallow networks; DCLNs can be much better (for compositional functions) but not for all functions/computations

• Binarization leads to consider sparse Boolean function
When and why are deep networks better than shallow networks?

Why does SGD work so well for deep networks?
Parenthetical comment on i-theory

• Convolution and pooling in multilayer networks reduces *sample complexity*

• Theorems about extending invariance beyond position invariance and how to learn it from the environment

Anselmi and Poggio, 2016, MIT Press