# 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

CENTER FOR
Brains
Minds+
Machines

# Engineering of Intelligence: recent successes

# Intelligence: engineering



PIENSE
THINK
THINK
$3,400
$1,200

# Recent progress in AI


Home World Companies Markets Global Economy Lex Comment Management Life & Arts

PERSON IN THE NEWS

# Demis Hassabis, master of the new machine age

Murad Ahmed

Share Author alerts Print Clip Comments

**The creator of the AI game-playing program makes all the right moves, writes Murad Ahmed**


Google DeepMind
Challenge Match
AlphaGo
KBA
Google

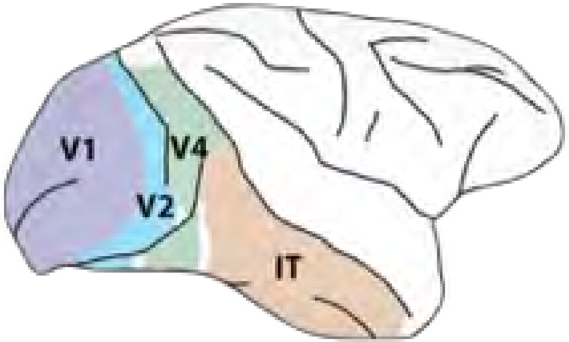
# Why now: very recent progress in AI

# Mobileye

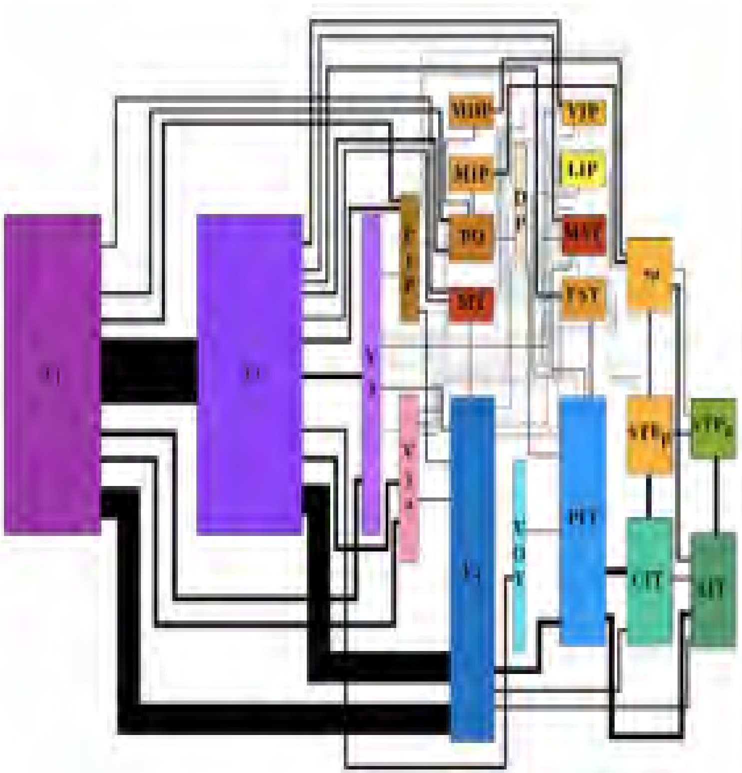
# 20 years ago: MIT and Daimler

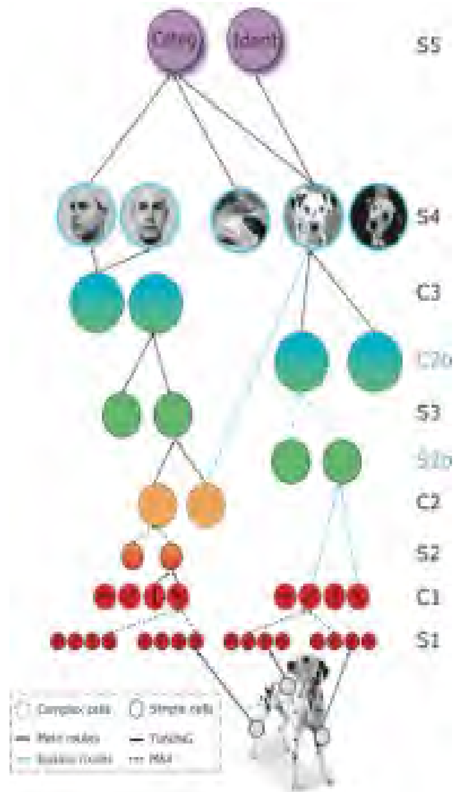
# CBMM: motivations

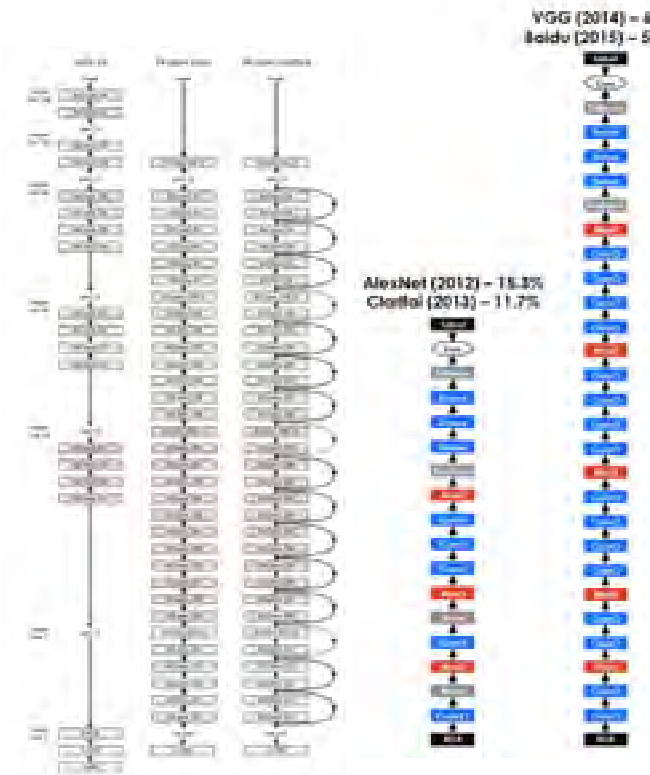
Key recent advances
in the engineering of intelligence
have their roots
in basic science of the brain




# The same hierarchical architectures in the cortex, in models of vision and in Deep Learning networks




Desimone & Ungerleider 1989; vanEssen+Movshon

CENTER FOR
Brains
Minds+
Machines

# The race for Intelligence

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

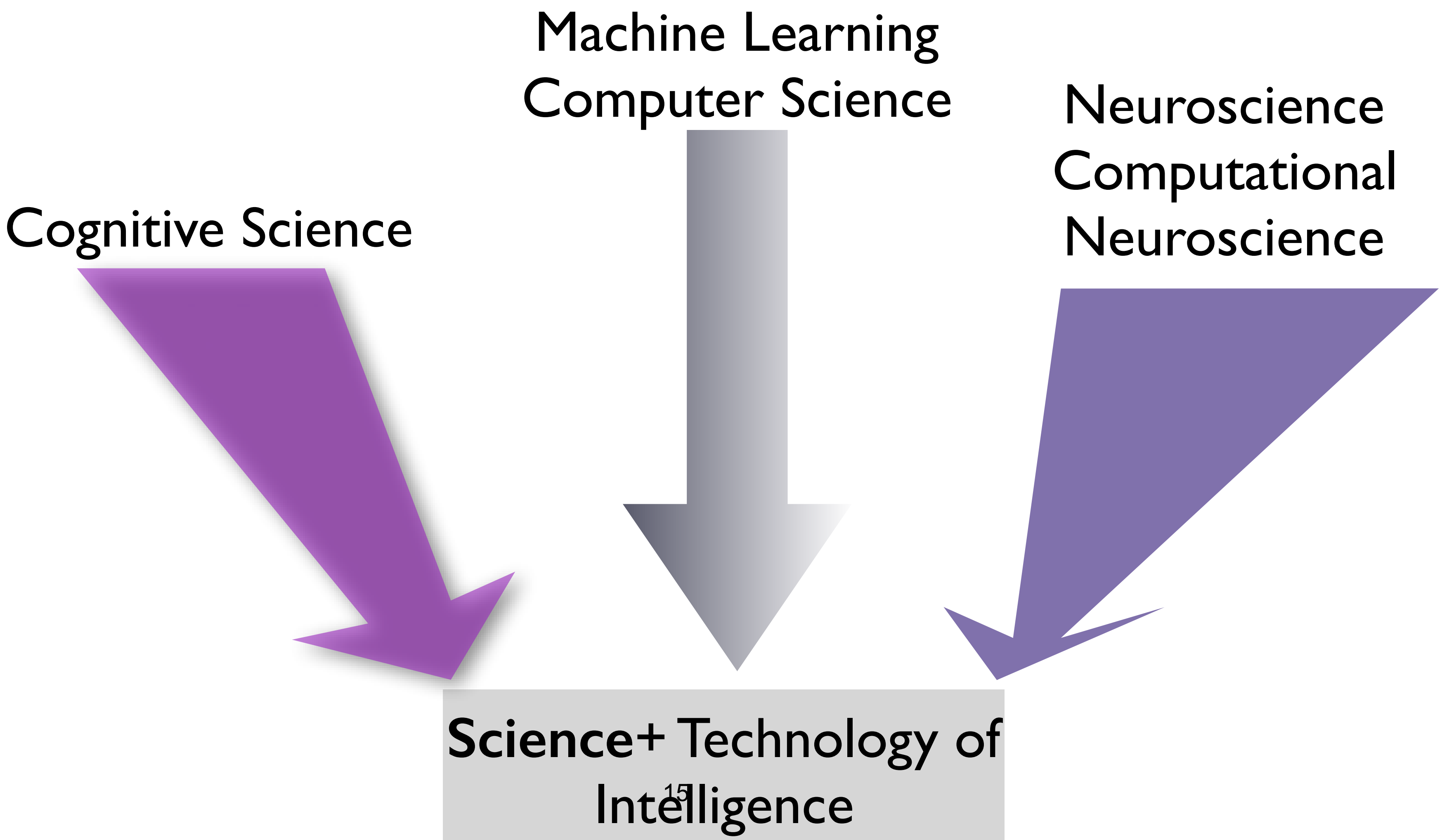
- …we need to make another basic effort on it
  - for the sake of *basic science*
  - for the *engineering of tomorrow*

# Science + Engineering of Intelligence

**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 <u>main</u> goal is to *make progress in the science of intelligence* which *enables better engineering* of intelligence.**

# Interdisciplinary



# 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

MEXT, Japan

**City U. HK**
Smale

**Google**

**IBM**

**Microsoft**

**Siemens**

**Schlumberger**

**GE**

**DeepMind**

Honda

**Boston Dynamics**

**Orcam**

Nvidia

**Rethink Robotics**

**MobilEye**

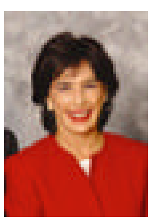
# EAC members

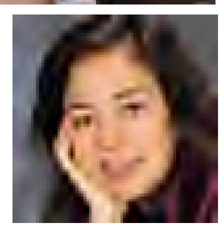


Pietro Perona, Caltech
Charles Isbell, Jr., Georgia Tech
Joel Oppenheim, NYU


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
Machines

# 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*:**

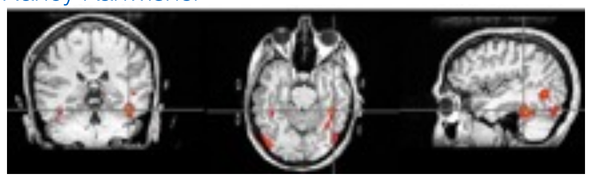
| | | | |
|---|---|---|---|
| Tomaso Poggio | Gabriel Kreiman | Nancy Kanwisher | Boris Katz |
| Winrich Freiwald | Matthew Wilson | Josh Tenenbaum | L Mahadevan |
| Elizabeth Spelke | Rebecca Saxe | Shimon Ullman | Laura Schulz |
| Ken Nakayama | Patrick Winston | Lorenzo Rosasco | Ethan Meyers |
|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

* CBMM faculty, industrial partners

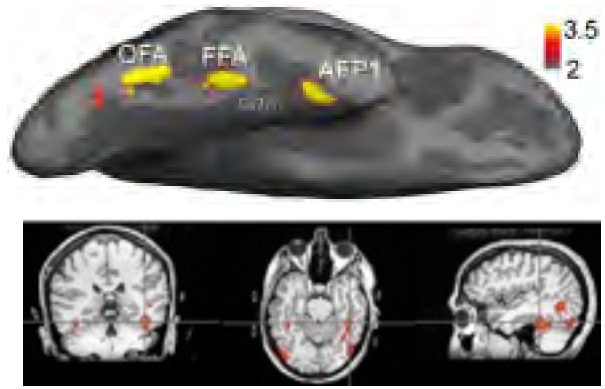
# Learning by Doing:
# Lab Work & Joint Student Projects

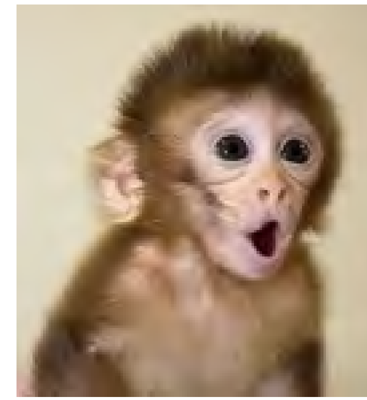
# An example project across thrusts: face recognition


Nancy Kanwisher

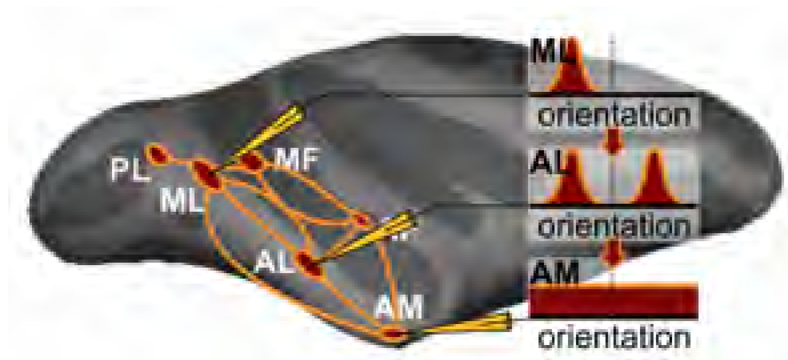


# A project across thrusts: face recognition

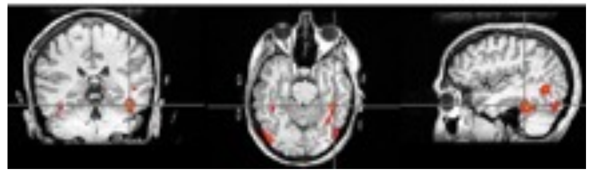


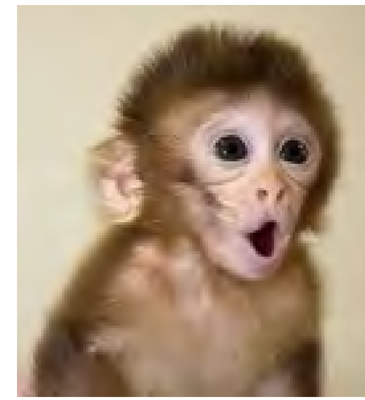



Winrich Freiwald and Doris Tsao



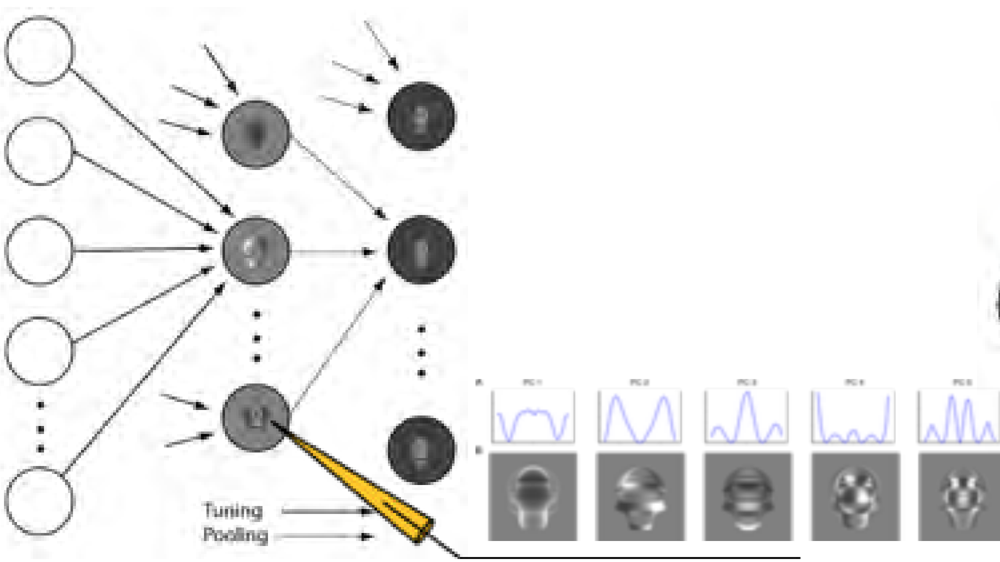
# A project across thrusts: face recognition

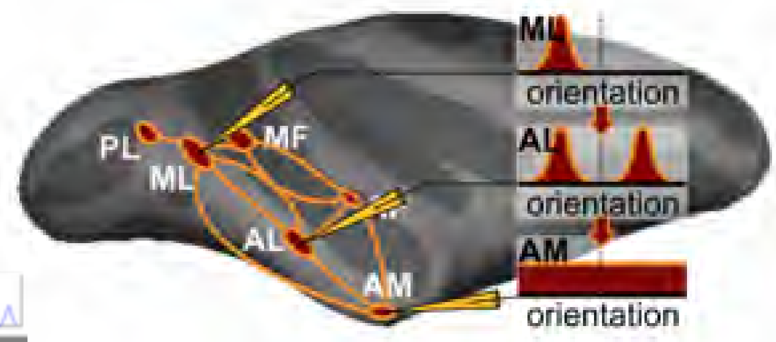





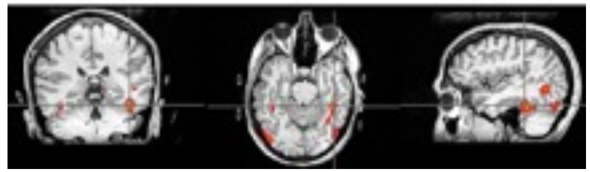
**Model**

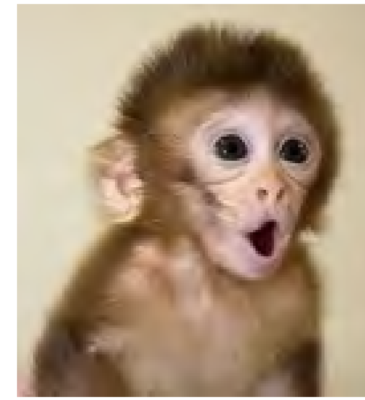
**ML** **AL** **AM**



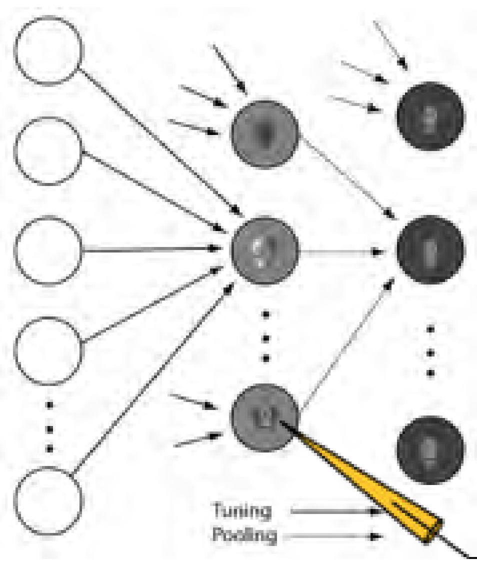
# A project across thrusts: face recognition

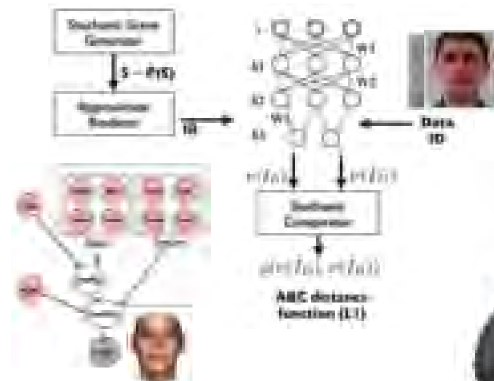



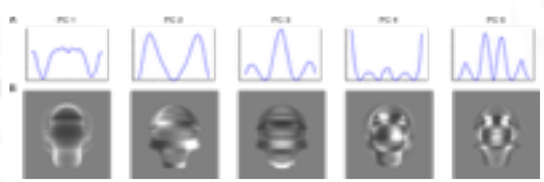


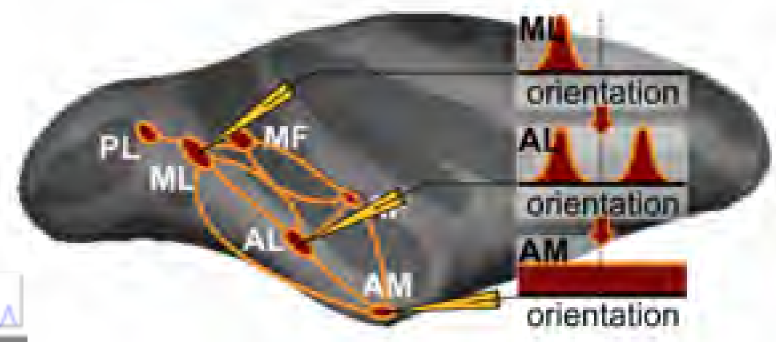
**Model**

**ML** **AL** **AM**







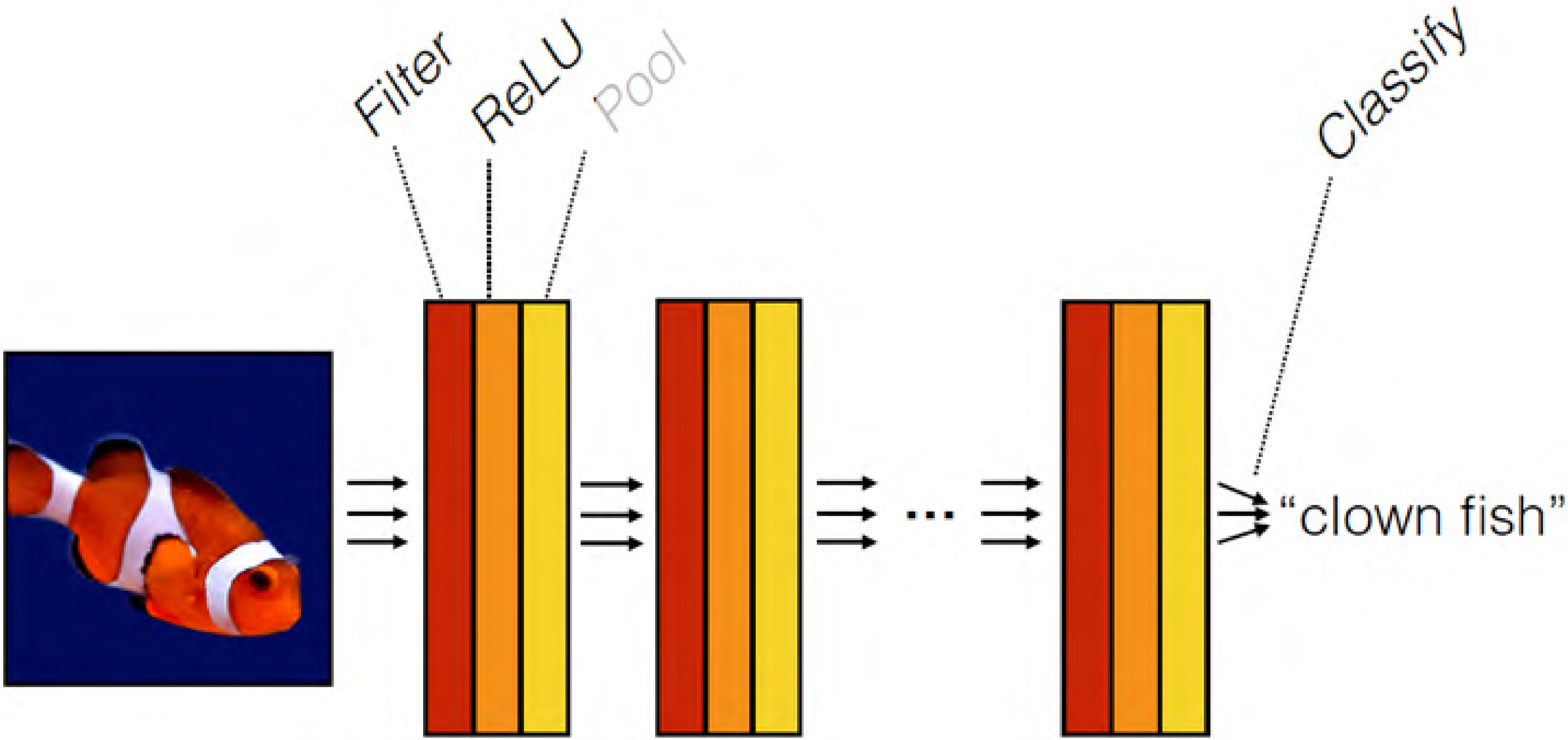

CENTER FOR
Brains
Minds+
Machines

# Another project

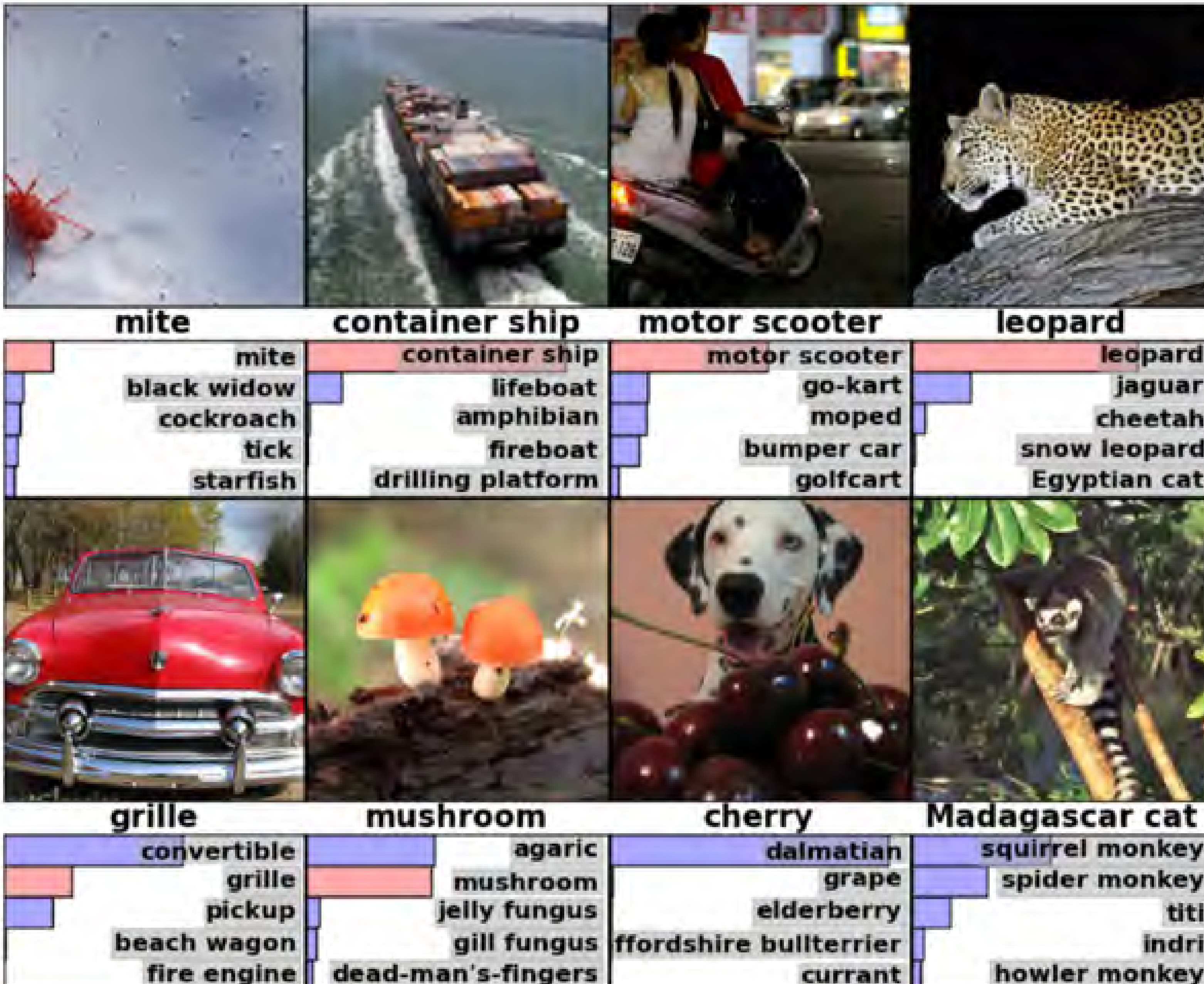
When and why are deep networks better than shallow networks?

Work with Hrushikeshl Mhaskar;
initial parts with L. Rosasco and F. Anselmi

# Computation in a neural net

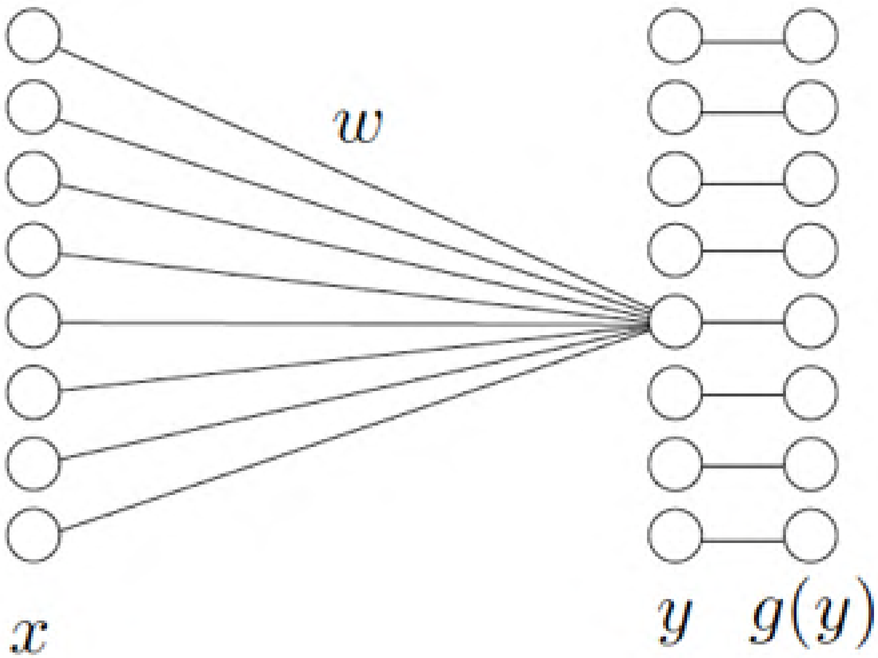


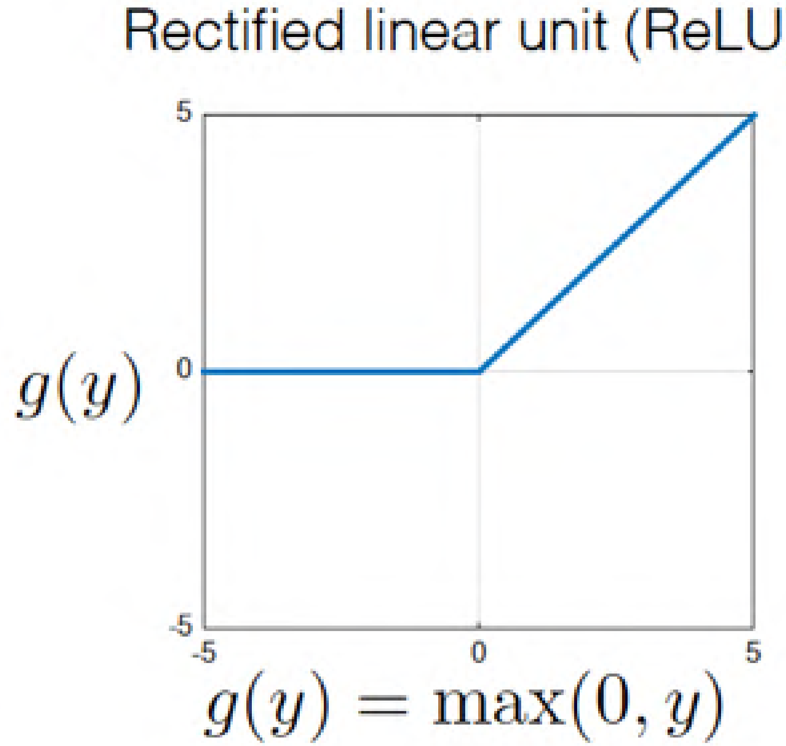

$$f(\mathbf{x}) = f_L(\dots f_2(f_1(\mathbf{x})))$$




Krizhevsky et al. NIPS 2012

# Computation in a neural net



# Gradient descent

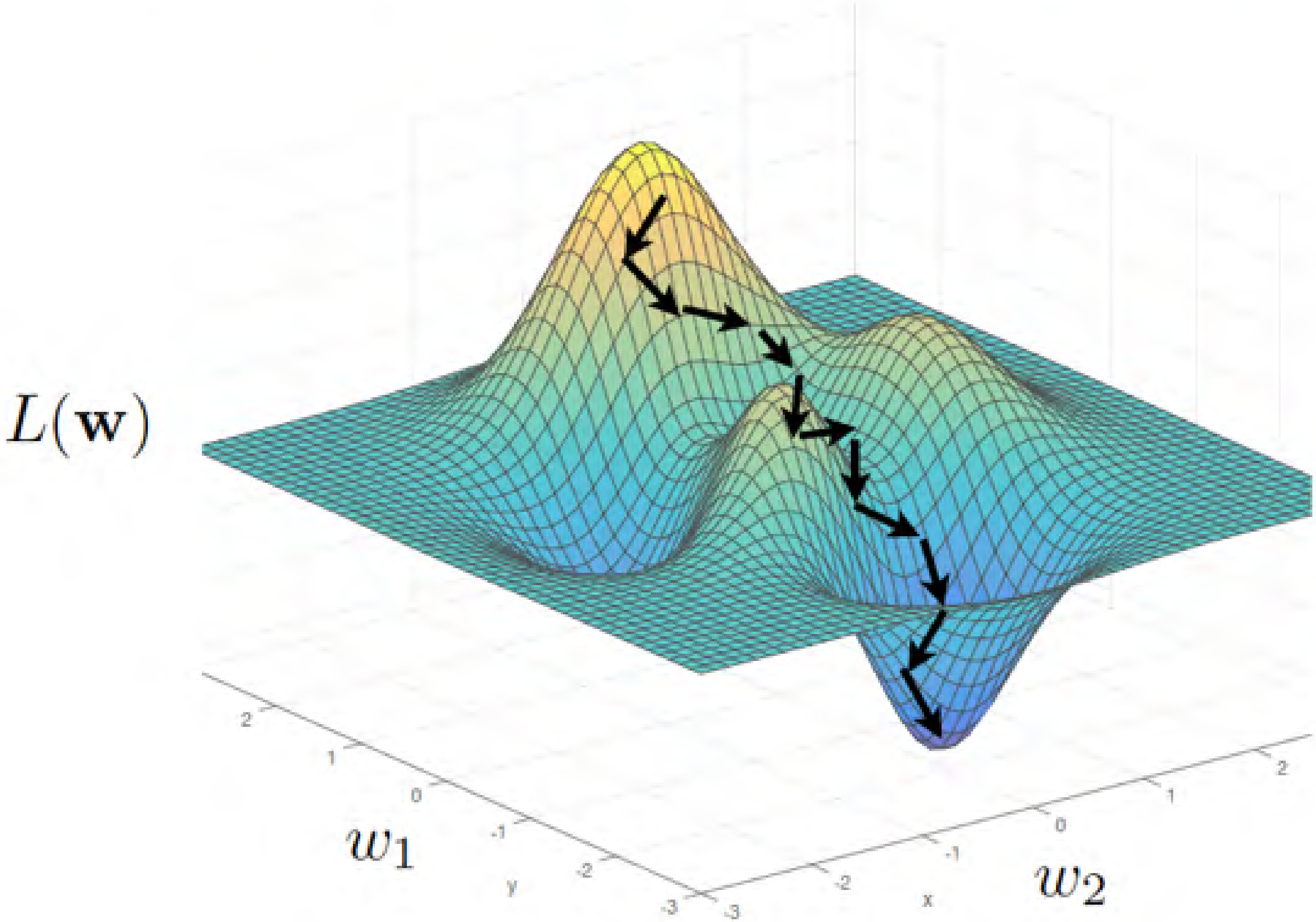
$$\boxed{\underset{\mathbf{w}}{\operatorname{argmin}} \quad \sum_i \ell(\mathbf{z}_i, f(\mathbf{x}_i; \mathbf{w})) = L(\mathbf{w})}$$

One iteration of gradient descent:

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

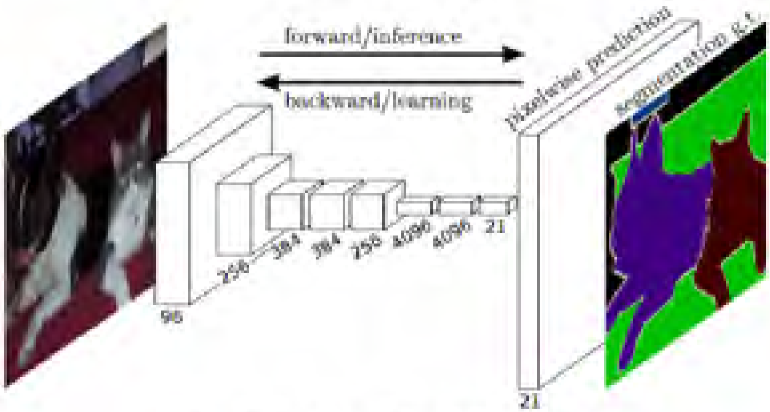
learning rate

# Stochastic gradient descent



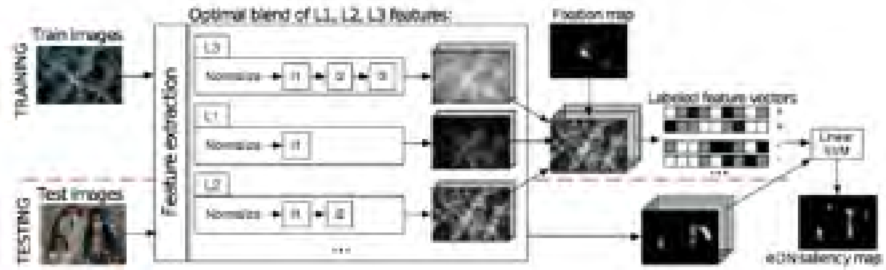
# What can you do with them?

## Parse images



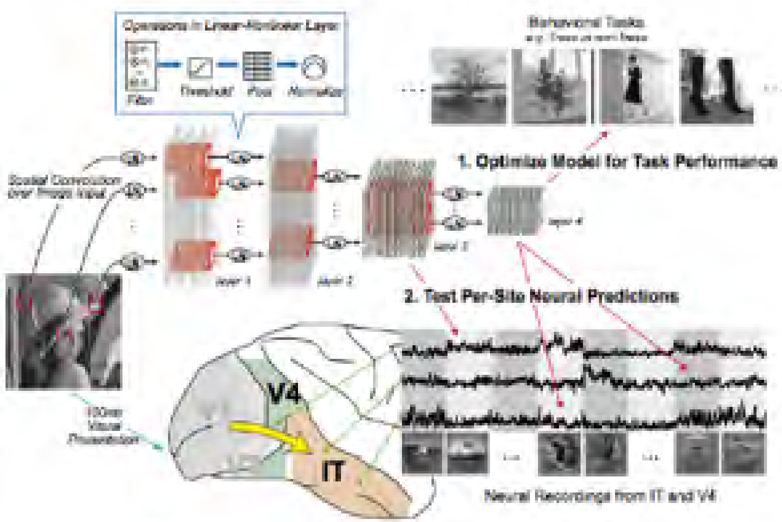

Long et al., CVPR 2015

## Model perception



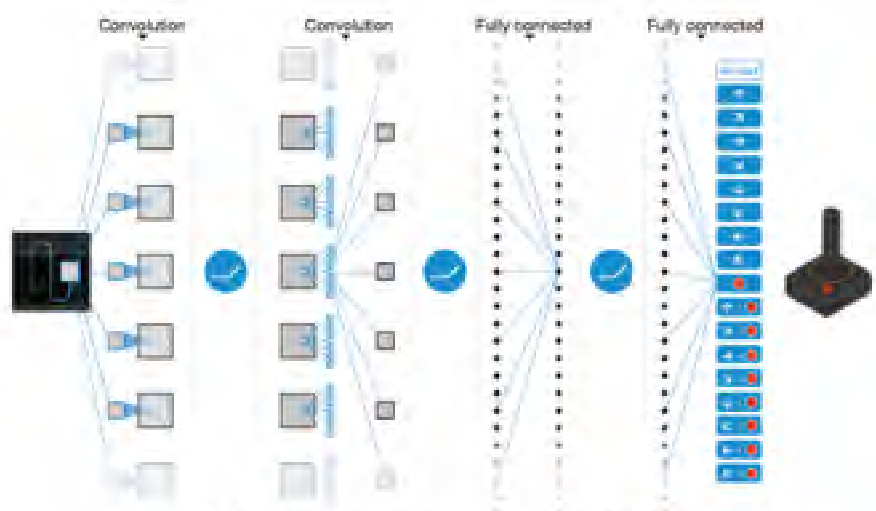

Vig et al., CVPR 2014

## Model the brain



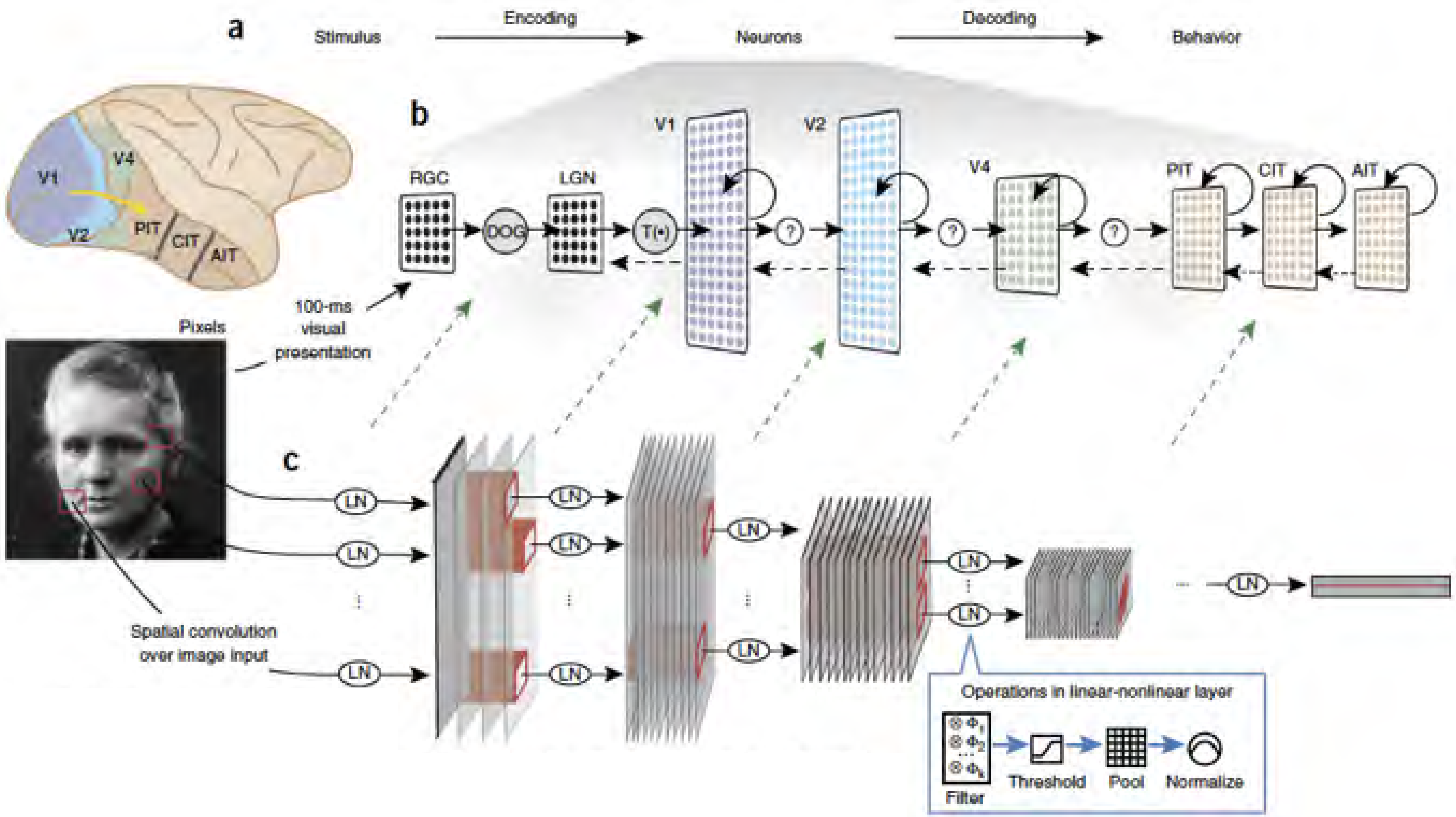

Yamins et al., PNAS 2014

## Beat humans at Atari games

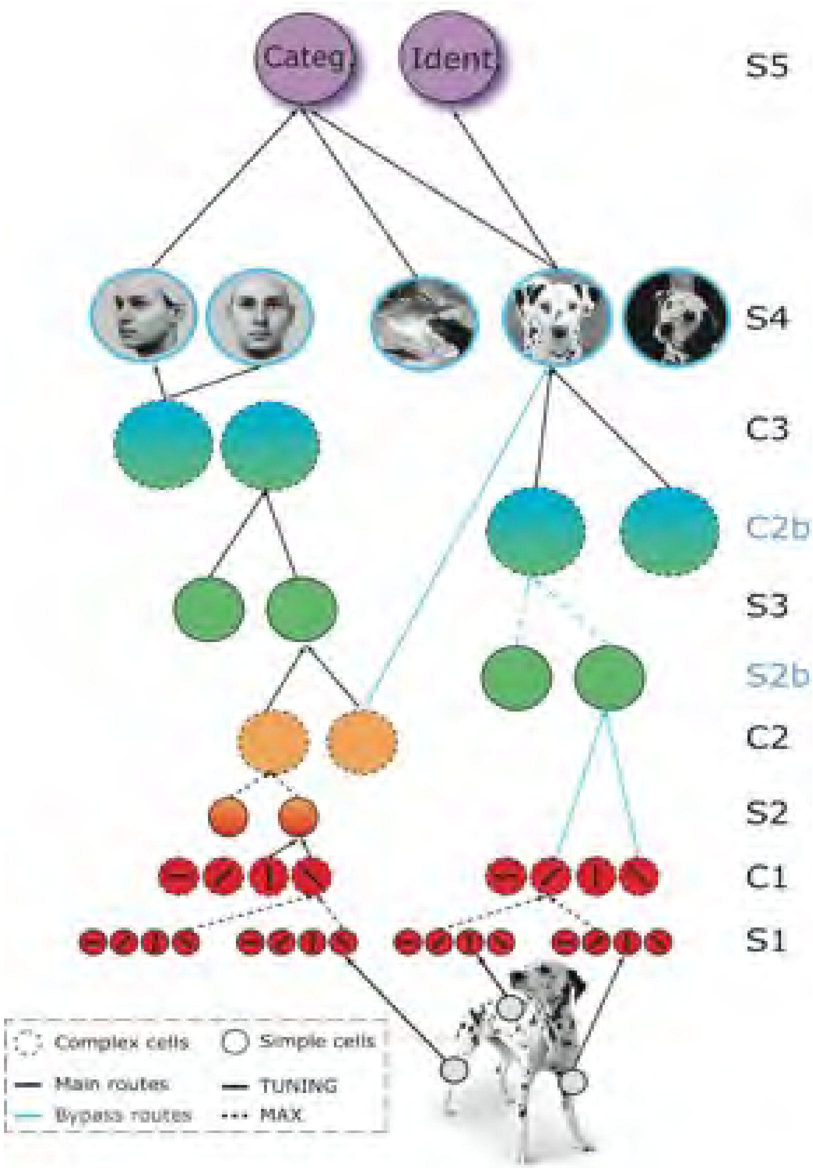



Mnih et al., Nature 2015

# Hierarchical feedforward models of the ventral stream do "work"



# **Convolutional** networks

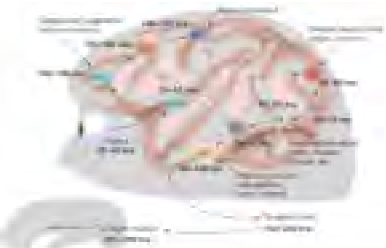


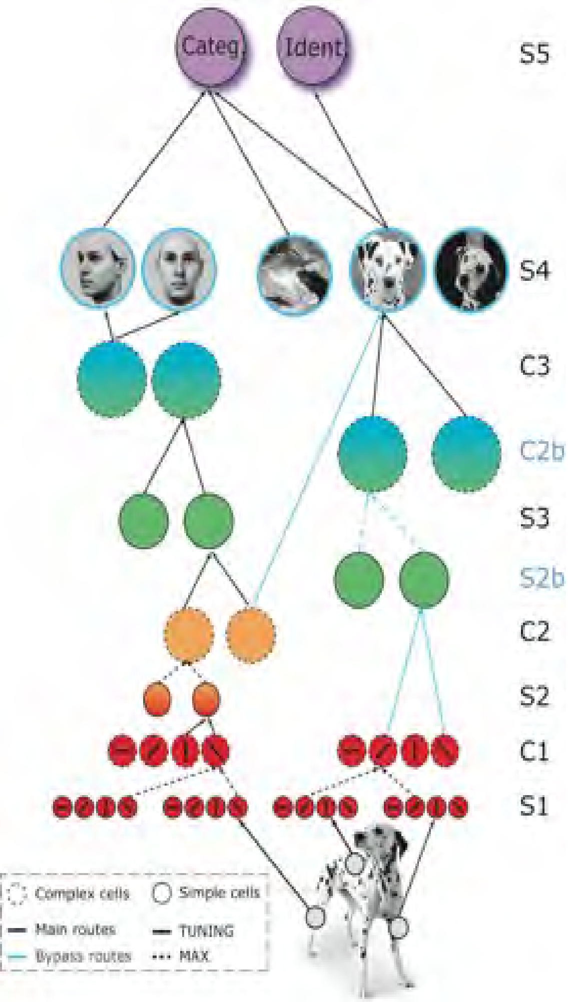

"Hubel-Wiesel" models include

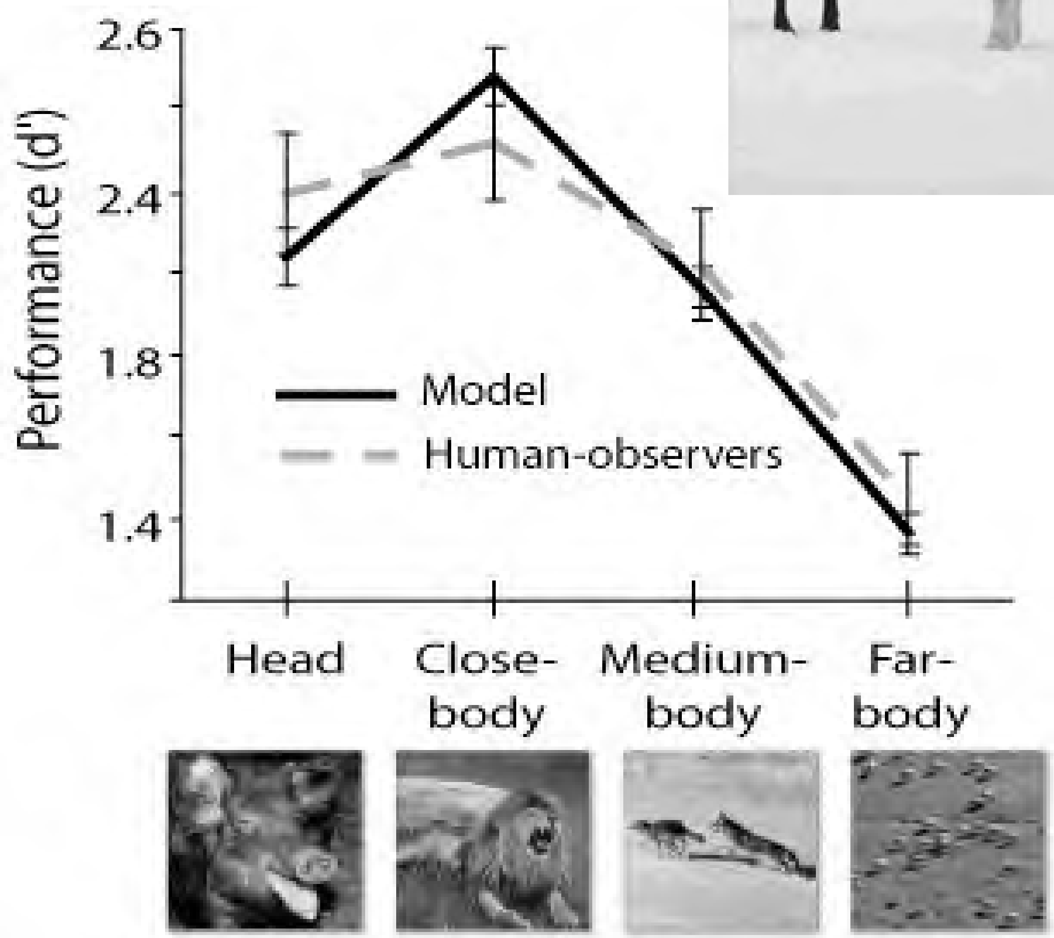
Hubel & Wiesel, 1959: Fukushima, 1980, Wallis & Rolls, 1997; Mel, 1997; LeCun et al 1998; Riesenhuber & Poggio, 1999; Thorpe, 2002; Ullman et al., 2002; Wersing and Koerner, 2003; Serre et al., 2007; Freeman and Simoncelli, 2011….

Riesenhuber & Poggio 1999, 2000; Serre Kouh Cadieu Knoblich Kreiman & Poggio 2005; Serre Oliva Poggio 2007

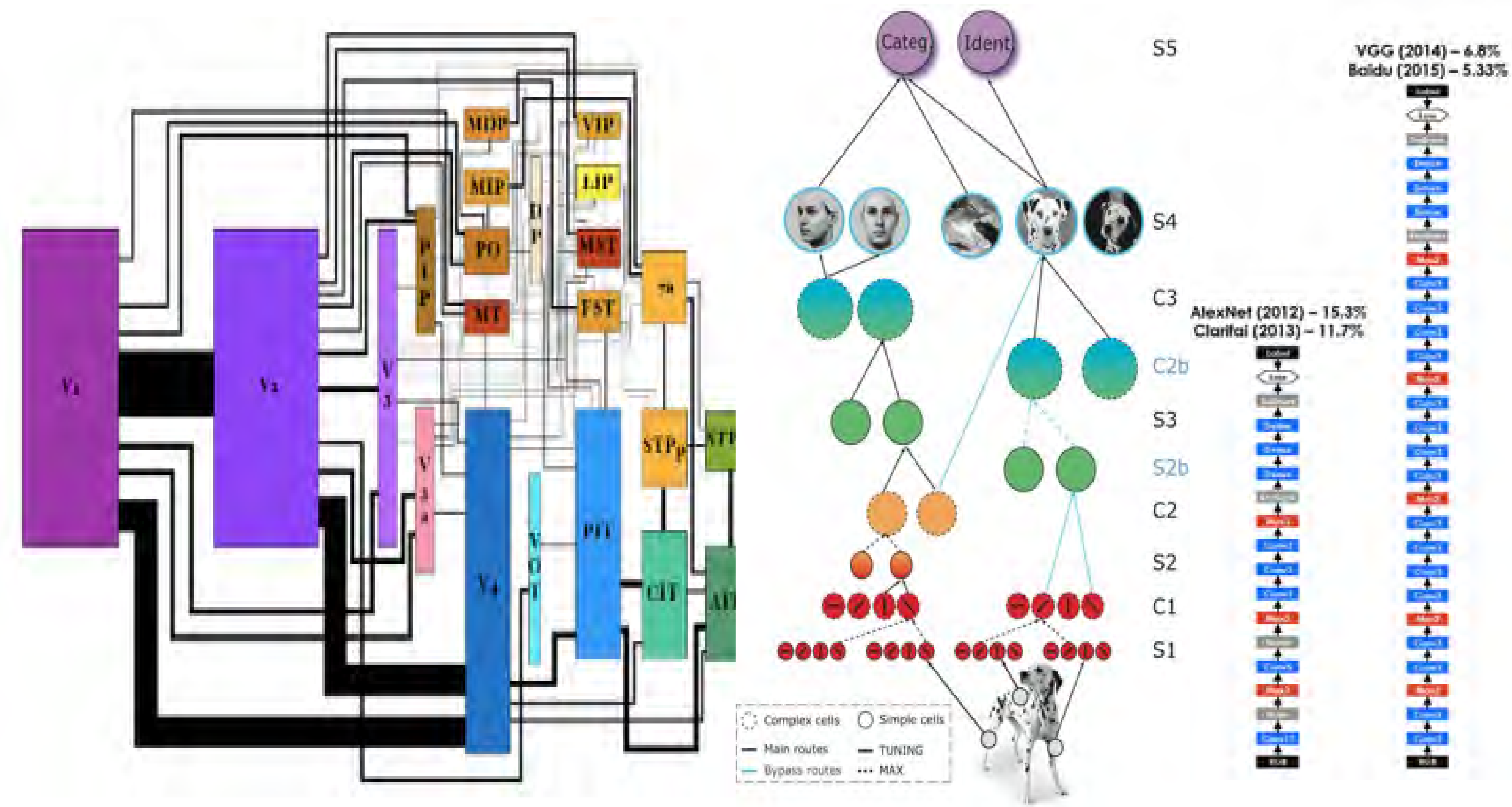
# Hierarchical feedforward models of the ventral stream do "work"





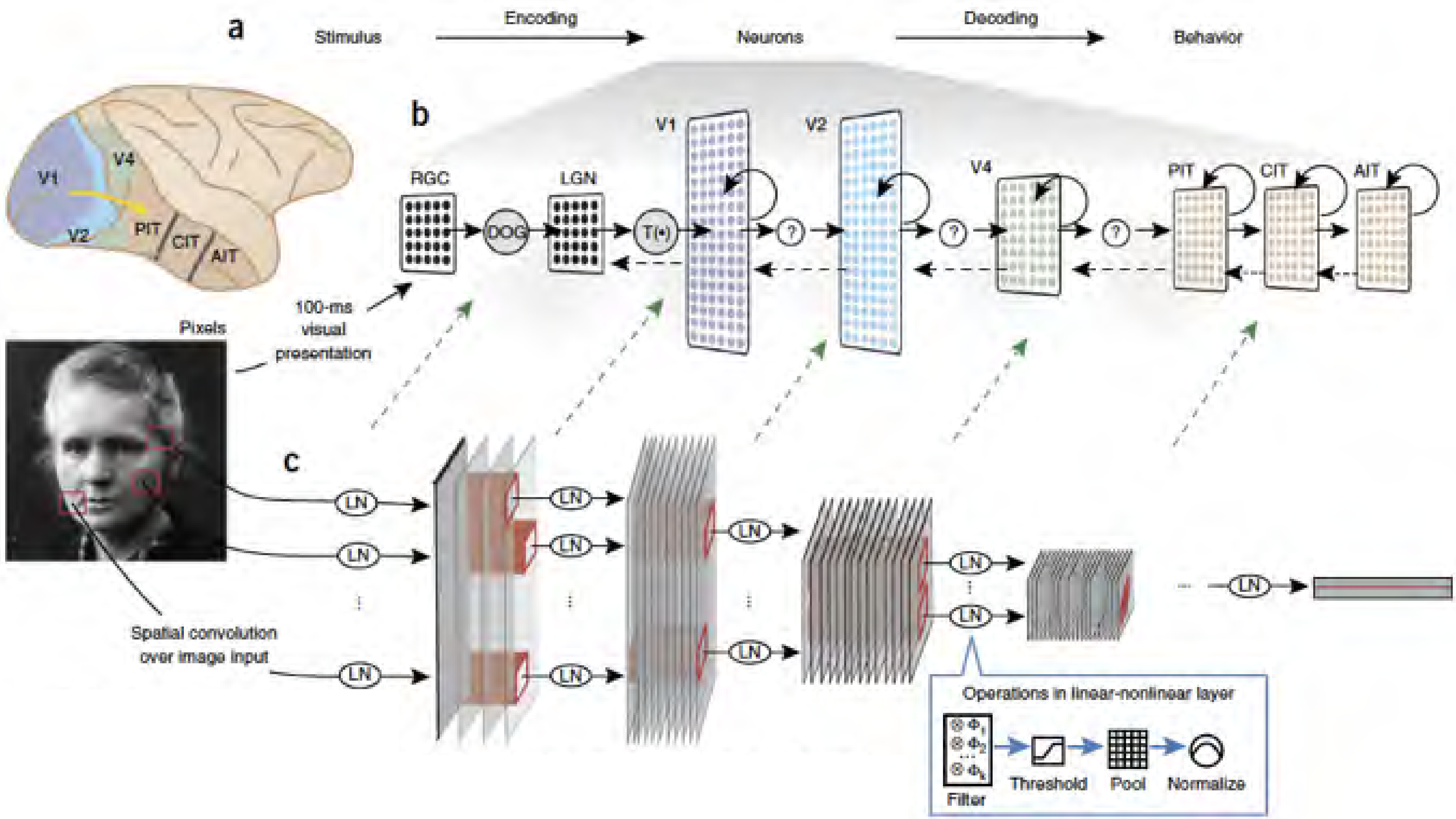



# The same hierarchical architectures in the cortex, in the models of vision and in Deep Learning networks



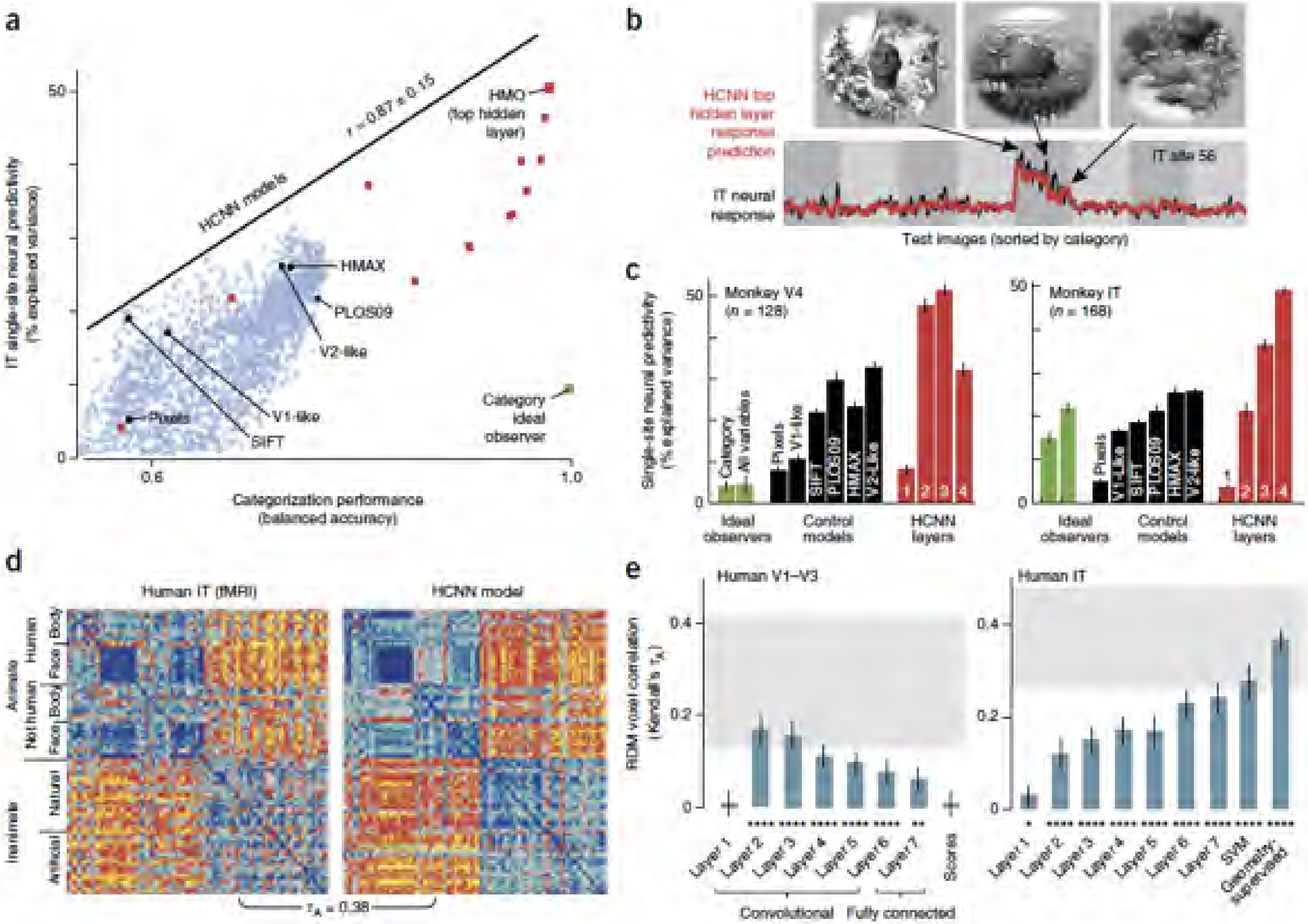
# Using goal-driven deep learning models to understand sensory cortex

Daniel L K Yamins[1,2] & James J DiCarlo[1,2]

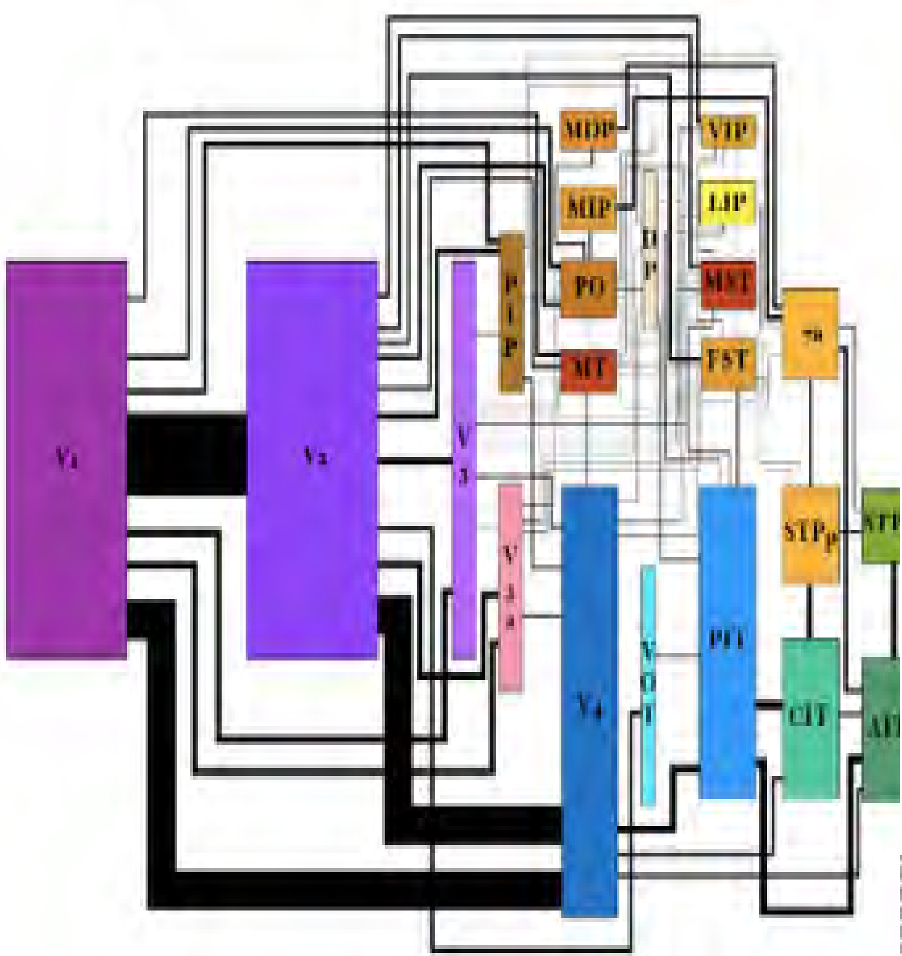


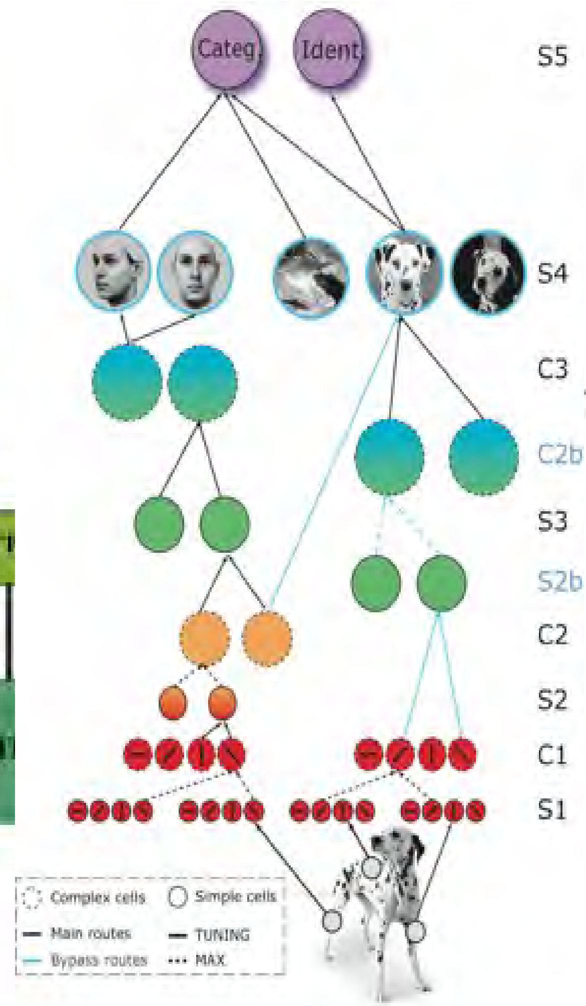
# Using goal-driven deep learning models to understand sensory cortex

Daniel L K Yamins[1,2] & James J DiCarlo[1,2]



# 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

**Classical learning algorithms:**
**“high” sample complexity and shallow architectures**

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?

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

# 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.*

$$g(x) = \sum_{i=1}^{r} c_i \left| \langle w_i, x \rangle + b_i \right|_+$$


Cybenko, Girosi, ….

# 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

$$f(\mathbf{x}) = \sum_i^l \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

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

RKHS were explicitly introduced in learning theory by Girosi (1997), Vapnik (1998). Moody and Darken (1989), and Broomhead and Lowe (1988) introduced RBF to learning theory. Poggio and Girosi (1989) introduced Tikhonov regularization in learning theory and worked (implicitly) with RKHS. RKHS were used earlier in approximation theory (eg Parzen, 1952-1970, Wahba, 1990).

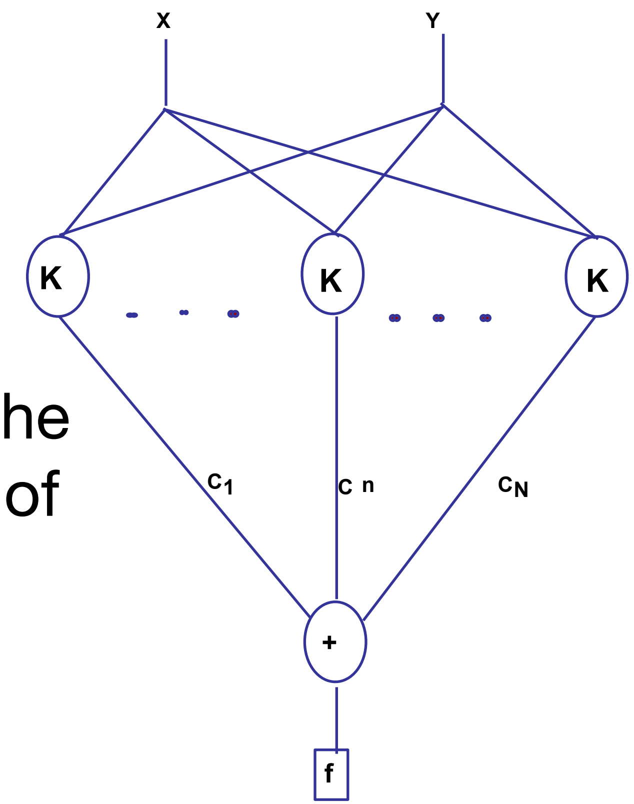
For a review, see Poggio and Smale, **The Mathematics of Learning**, Notices of the AMS, 2003

## Classical kernel machines are equivalent to shallow networks

Kernel machines…

$$f(\mathbf{x}) = \sum_i^l c_i K(\mathbf{x}, \mathbf{x}_i) + b$$

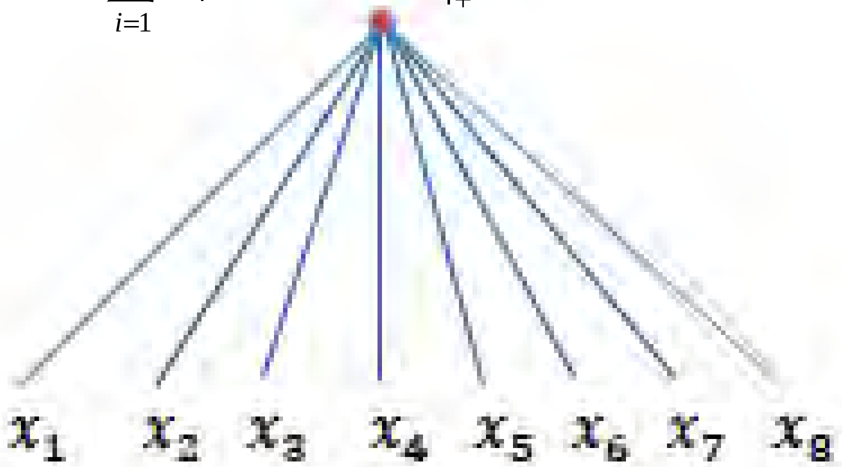
can be “written” as shallow networks: the value of K corresponds to the “activity” of the “unit” for the input and the correspond to “weights”

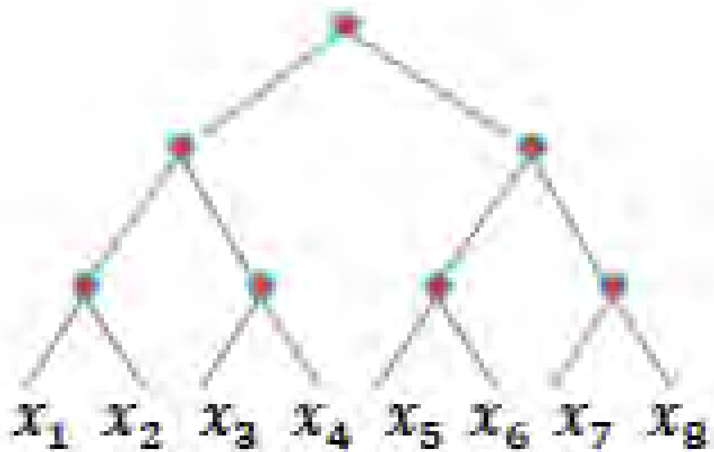


# Deep and shallow networks: universality

**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.*

$$g(x) = \sum_{i=1}^{r} c_i \left| \langle w_i, x \rangle + b_i \right|_+$$



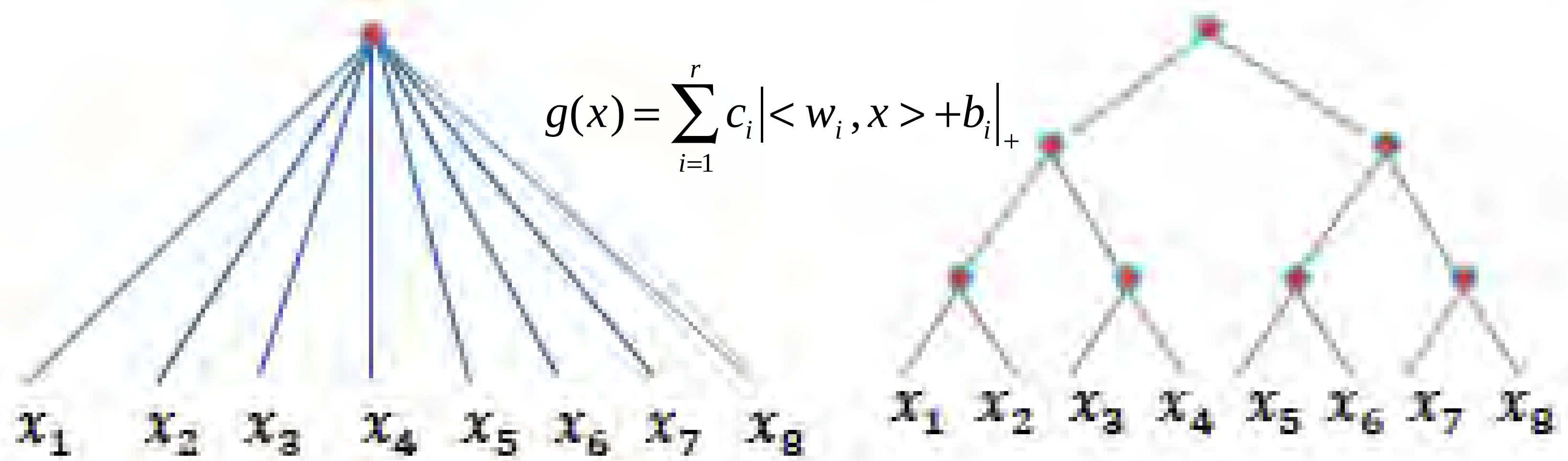




Cybenko, Girosi, ....

# Deep and shallow networks

- Thus depth is not needed to for approximation

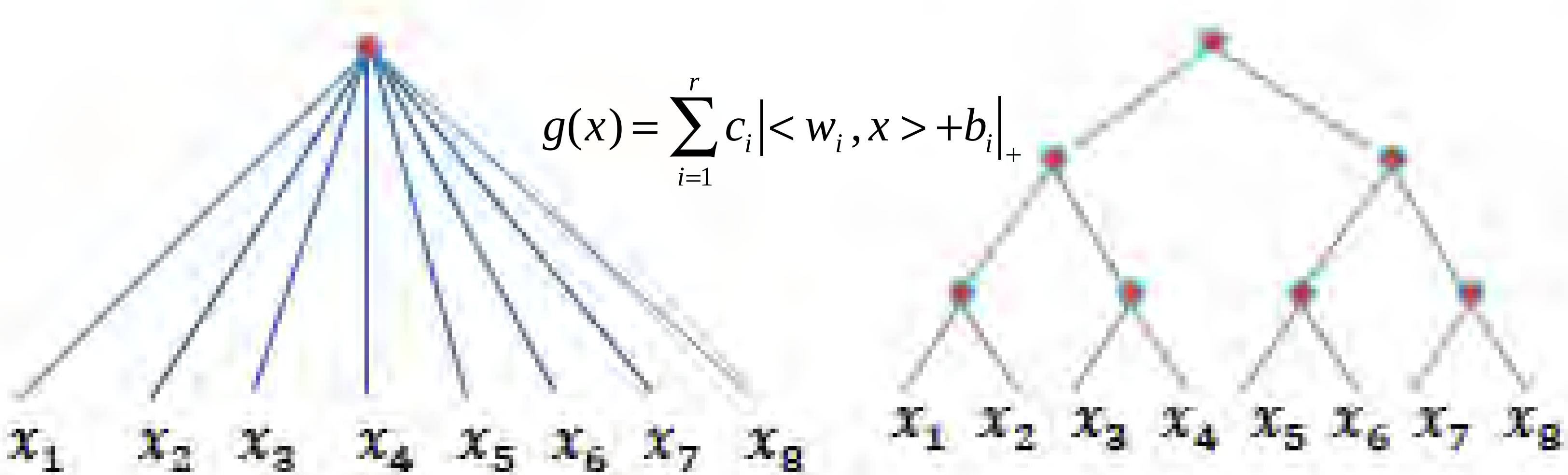
$$g(x) = \sum_{i=1}^{r} c_i \left| < w_i, x > + b_i \right|_+$$



# Deep and shallow networks

- Thus depth is not needed to for approximation

- Conjecture: depth may be more effective for certain classes of functions




$$g(x) = \sum_{i=1}^{r} c_i \left| < w_i, x > + b_i \right|_+$$

## Generic functions

$$f(x_1, x_2, ..., x_8)$$

## Compositional functions

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



Mhaskar, Poggio, Liao, 2016

# Theorem:
# why and when are deep networks better than shallow network?

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

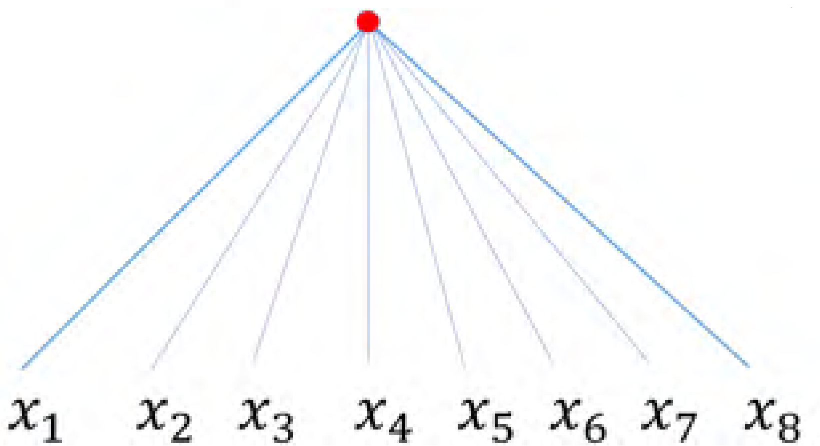
Center for Brains,
Minds & Machines

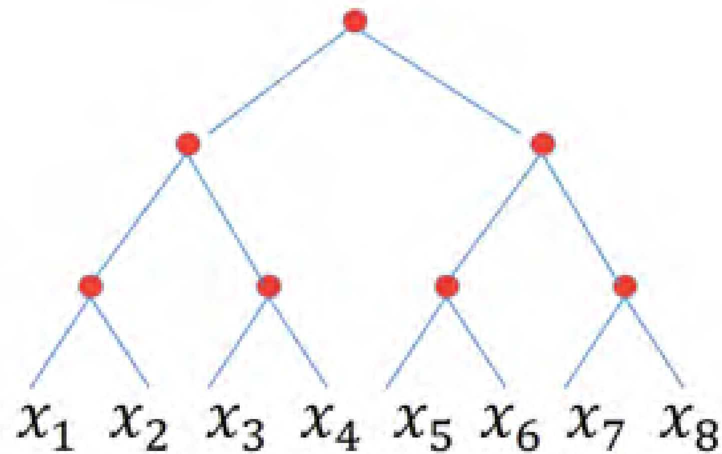
Mhaskar, Poggio, Liao, 2016

# Theorem:
## why and when are deep networks better than shallow network?

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

$$g(x) = \sum^{r} c_i \left| < w_i, x > + b_i \right|$$



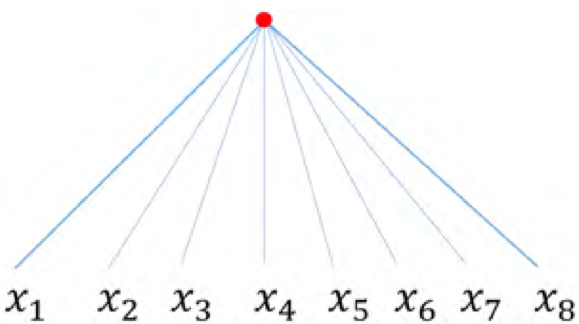



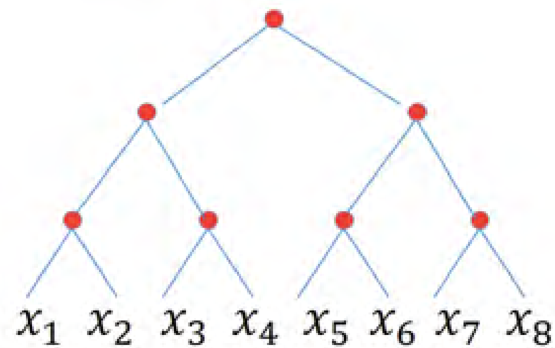

Center for Brains, Minds & Machines

Mhaskar, Poggio, Liao, 2016

# 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* $\mathbb{R}$.

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

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

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

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

This is the best possible estimate (n-width result)

Mhaskar, Poggio, Liao, 2016

# 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 6 k n 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))*

Center for Brains, Minds & Machines

Mhaskar, Poggio, Liao, 2016

# 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.

Center for Brains, Minds & Machines

Mhaskar, Poggio, Liao, 2016

# 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 $\mathbb{V}$, $\mathbb{W}$ be compact subsets of a metric space $\mathbb{X}$, and $d_n(\mathbb{V})$ (respectively, $d_n(\mathbb{W})$) be their $n$-widths. We define the *relative dimension* of $d_n(\mathbb{V})$ with respect to $d_n(\mathbb{W})$ by

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

Thus, $D(\mathcal{GB}_\gamma, \mathcal{B}_{\gamma,q}) \leq d/q$. This leads us to say that $\mathbb{V}$ is *parsimonious* with respect to $\mathbb{W}$ if $D(\mathbb{V}, \mathbb{W}) \ll 1$.

# 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

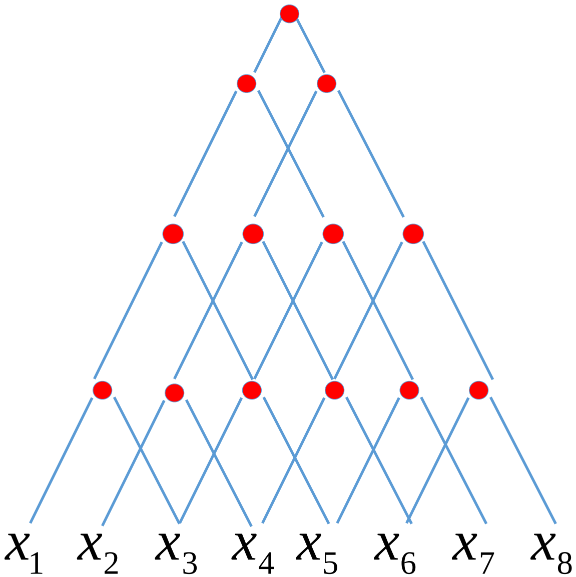
1. A binary tree net is a good proxy for ResNets

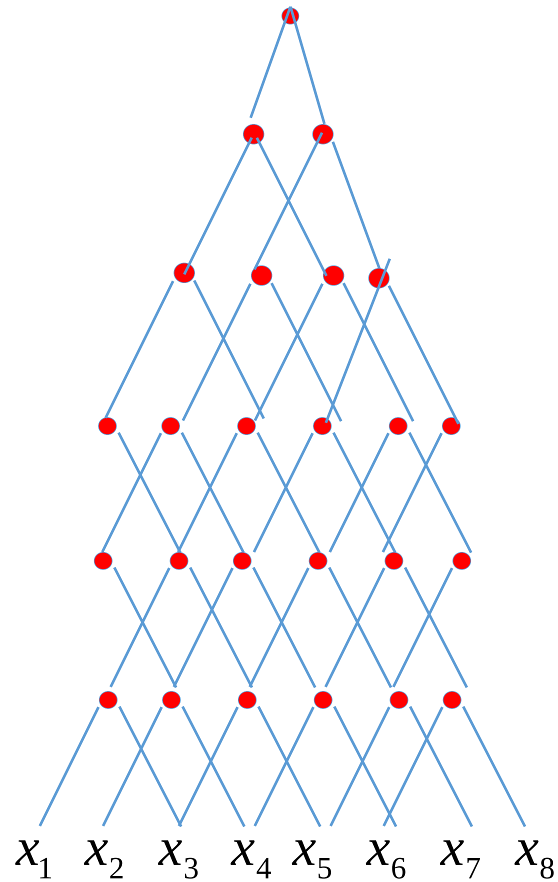
2. Scalable algorithms and compositional functions

4. Invariance and pooling

6. Sparse functions and Boolean functions

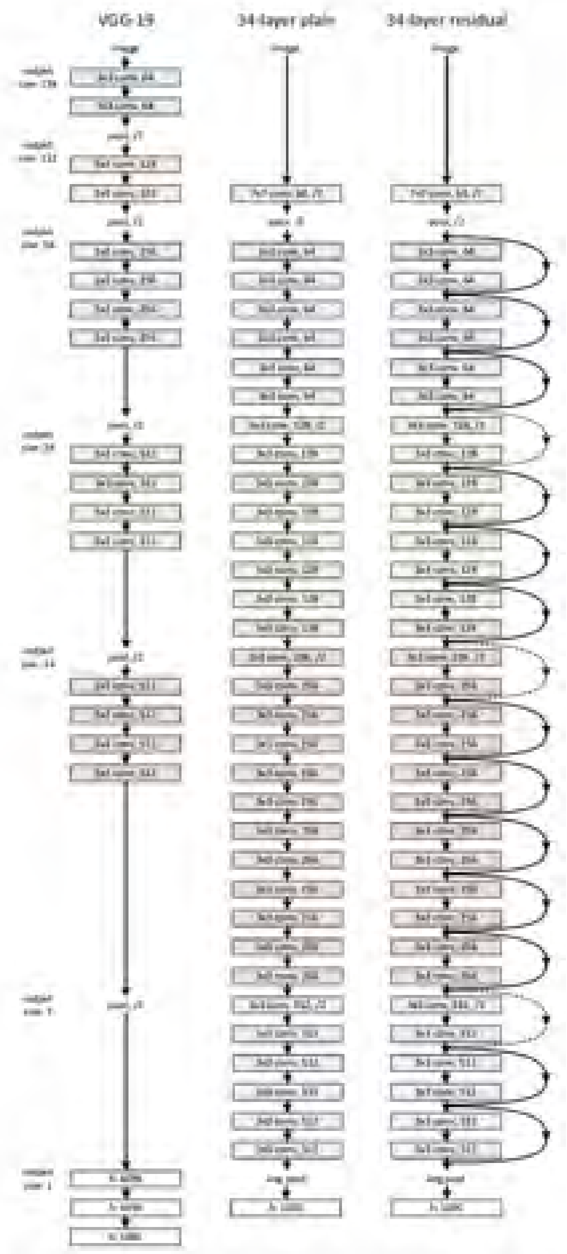
# Convolutional Deep Networks (no pooling like in ResNets))



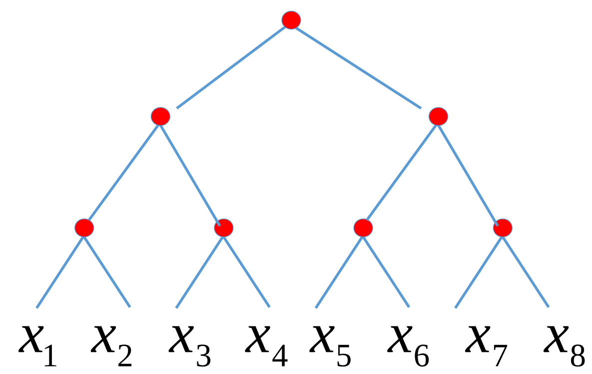




Similar theorems apply to the network on the left and the network on the right in terms of # parameters

# Hyper deep residual networks:
# a binary tree net is a good mathematical proxy




$\sim$



# Remarks

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

2. Scalable algorithms and compositional functions

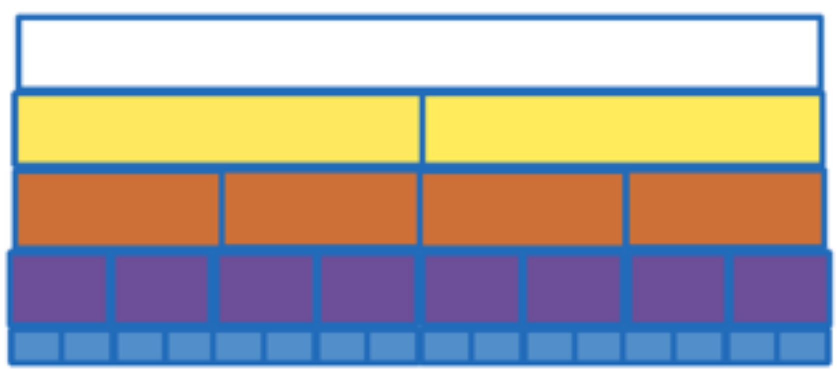
4. Invariance and pooling

6. Sparse functions and Boolean functions

# Shift-invariant, scalable algorithms

**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 $K_{2M} : \mathbb{R}^{2M} \to \mathbb{R}$ 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.



Mhaskar, Poggio, Liao, 2016

## 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 s 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)



Anselmi and Poggio, 2016, MIT Press

# Remarks

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

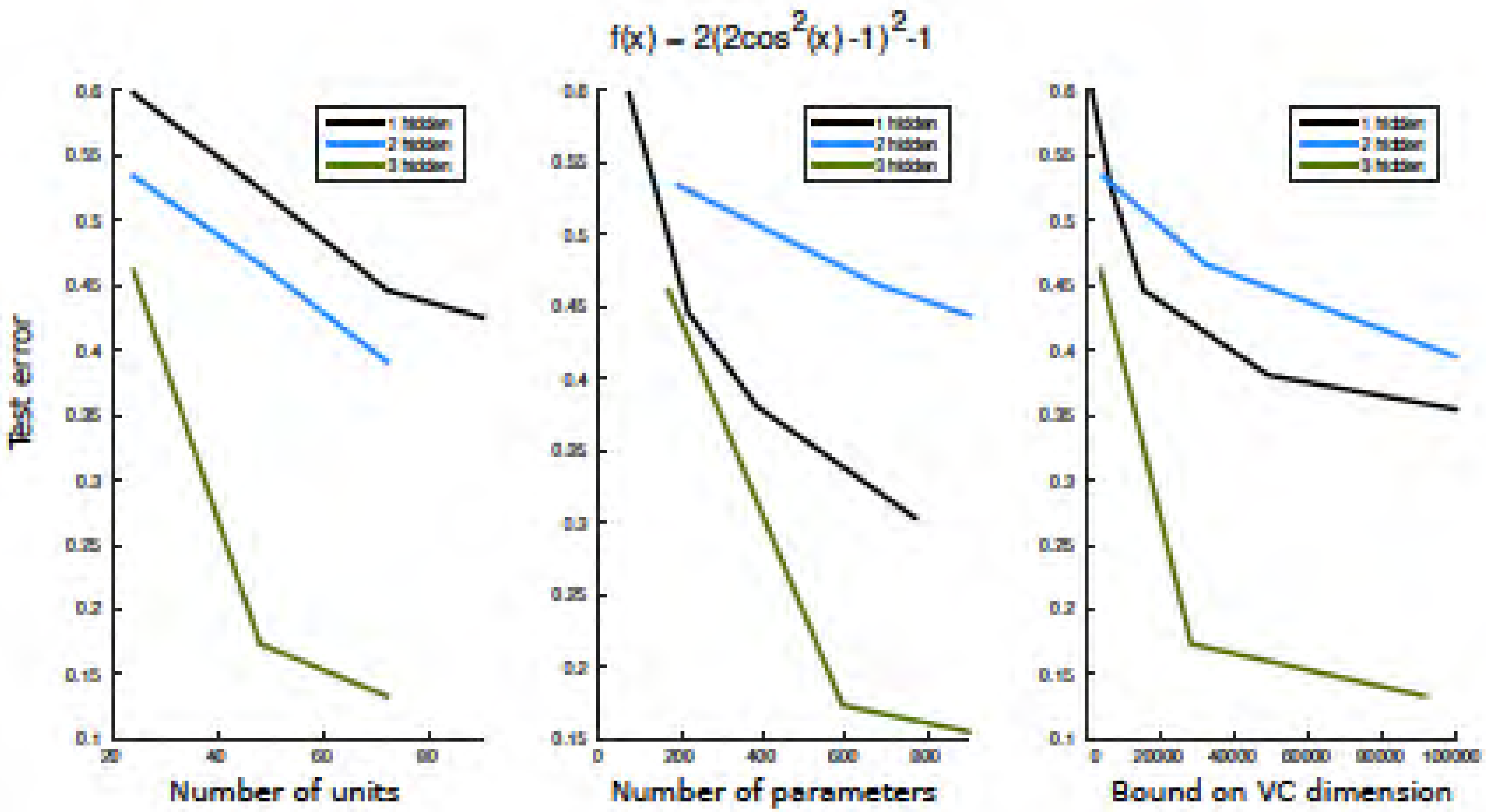
2.Scalable algorithms and compositional functions

4. Invariance and pooling

6. Sparse functions and Boolean functions

# Sparse functions

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




Mhaskar, Poggio, Liao, 2016

# 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

# DLNNs: two main scientific questions

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