Statistical Learning Theory and Applications

9.520/6.860 in Fall 2018

Class Times:
Tuesday and Thursday 11am-12:30pm in 46-3002 Singleton Auditorium
Units: 3-0-9 H,G
Web site: http://www.mit.edu/~9.520/fall19/

Contact: 9.520@mit.edu

Tomaso Poggio (TP), Lorenzo Rosasco (LR), Sasha Rakhlin (SR)

TAs:
Andrzej Banburski, ,
Michael Lee

Qianli Liao

Rules of the game

Today's overview

- Course description/logistic
- Motivations for this course: a golden age for Machine Learning, CBMM, MIT: ***Intelligence, the Grand Vision***
- A bit of history: Statistical Learning Theory, Neuroscience
- A bit of ML history: applications
- Deep Learning present and future

9.520: Statistical Learning Theory and Applications

Course focuses on algorithms and theory for supervised learning — **no applications**!

1. Classical regularization (regularized least squares, SVM, logistic regression, square and exponential loss), stochastic gradient methods, implicit regularization and minimum norm solutions. Regularization techniques, Kernel machines, batch and online supervised learning, sparsity.

2. Classical concepts like generalization, uniform convergence and Rademacher complexities will be developed, together with topics such as surrogate loss functions for classification, bounds based on margin, stability, and privacy.

3. Theoretical frameworks addressing three key puzzles in deep learning: approximation theory -- which functions can be represented more efficiently by deep networks than shallow networks-- optimization theory -- why can stochastic gradient descent easily find global minima -- and machine learning -- how generalization ideep networks used for classification can be explained in terms of complexity control implicit in gradient descent. It will also discconnections with the architecture of the brain, which was the originalinspiration of the layered local connectivity of modern networks and may provide ideas for future developments and revolutions in networks for learning.

9.520: Statistical Learning Theory and Applications

- Course focuses on algorithms and theory for supervised learning — **no applications**!
  - Classical regularization (regularized least squares, SVM, logistic regression, square and exponential loss), stochastic gradient methods, implicit regularization and minimum norm solutions. Regularization techniques, kernel machines, batch and online supervised learning, sparsity.

9.520: Statistical Learning Theory and Applications

- Course focuses on algorithms and theory for supervised learning — **no applications**!

  - Classical concepts like generalization, uniform convergence and Rademacher complexities will be developed, together with topics such as surrogate loss functions for classification, bounds based on margin, stability, and privacy.

9.520: Statistical Learning Theory and Applications

- Course focuses on algorithms and theory for supervised learning — **no applications**!

- Theoretical frameworks addressing three key puzzles in deep learning: approximation theory -- which functions can be represented more efficiently by deep networks than shallow networks-- optimization theory -- why can stochastic gradient descent easily find global minima -- and machine learning -- how generalization ideep networks used for classification can be explained in terms of complexity control implicit in gradient descent. It will also discuss connections with the architecture of the brain, which was the original inspiration of the layered local connectivity of modern networks and may provide ideas for future developments and revolutions in networks for learning.

Today's overview

- Course description/logistic
- Motivations for this course: a golden age for new AI, the key role of Machine Learning, CBMM, the MIT Quest: ***Intelligence, the Grand Vision***
- Bits of history: Statistical Learning Theory, Neuroscience
- Bits of ML history: applications
- Deep Learning

Grand Vision of CBMM, Quest/College, this course

The problem of intelligence: how the brain creates intelligence and how to replicate it in machines

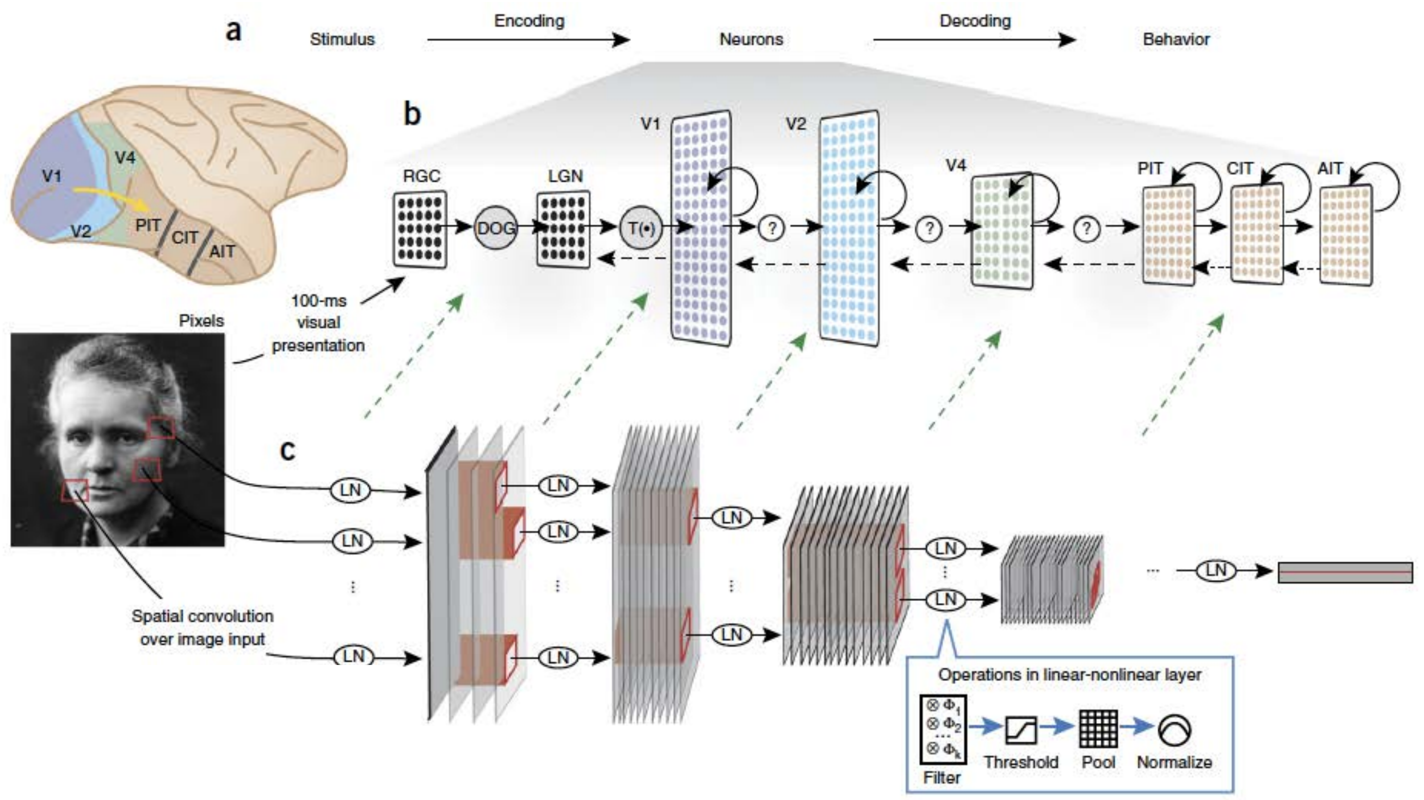
The problem of (human) intelligence is one of the great problems in science, probably the greatest.

Research on intelligence:

- a great intellectual mission: understand the brain, reproduce it in machines
- will help develop intelligent machines

The Science and the Engineering of Intelligence

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.

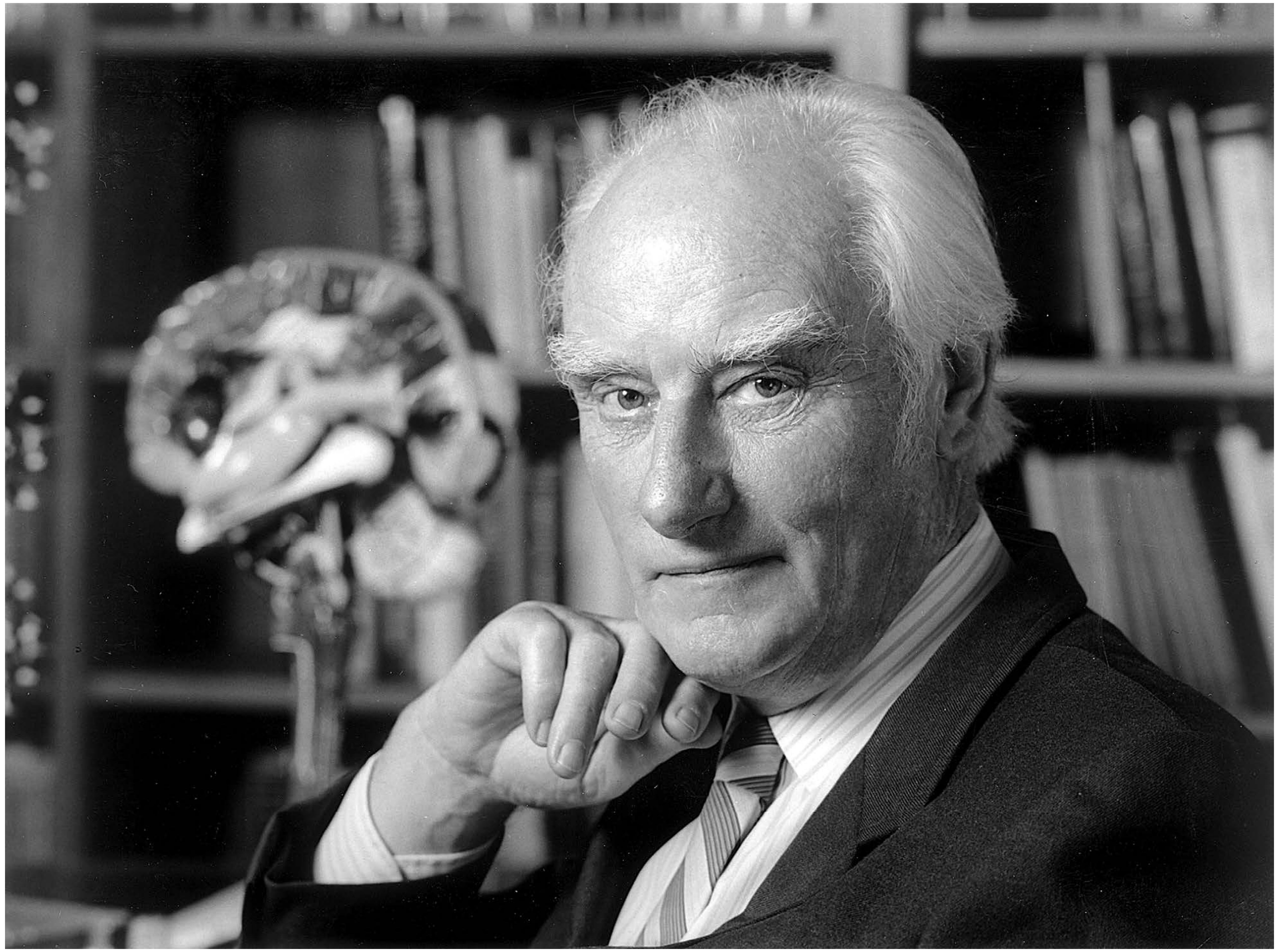


Key recent advances in the engineering of intelligence have their roots in basic research on the brain

CENTER FOR
Brains
Minds+
Machines

Why (Natural) Science
and
Engineering?

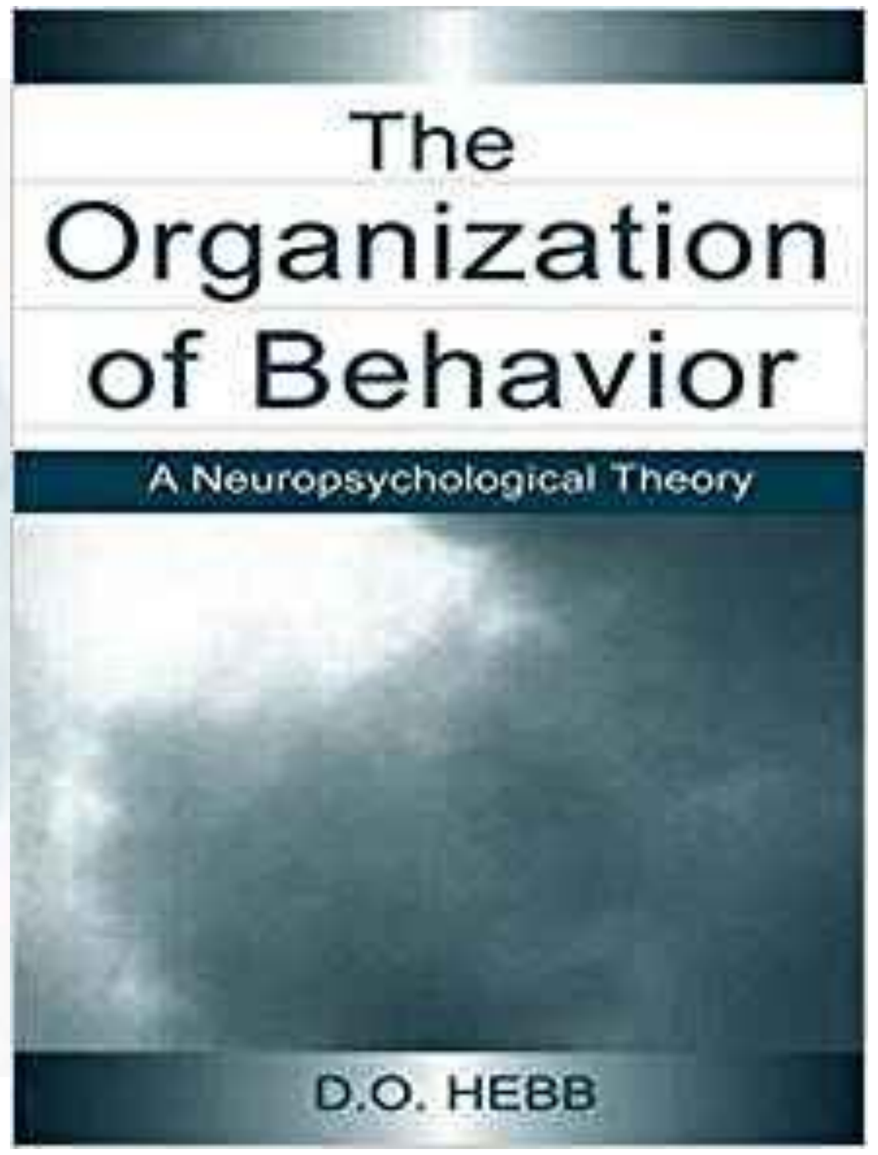
Just a definition: science is natural science (Francis Crick, 1916-2004)

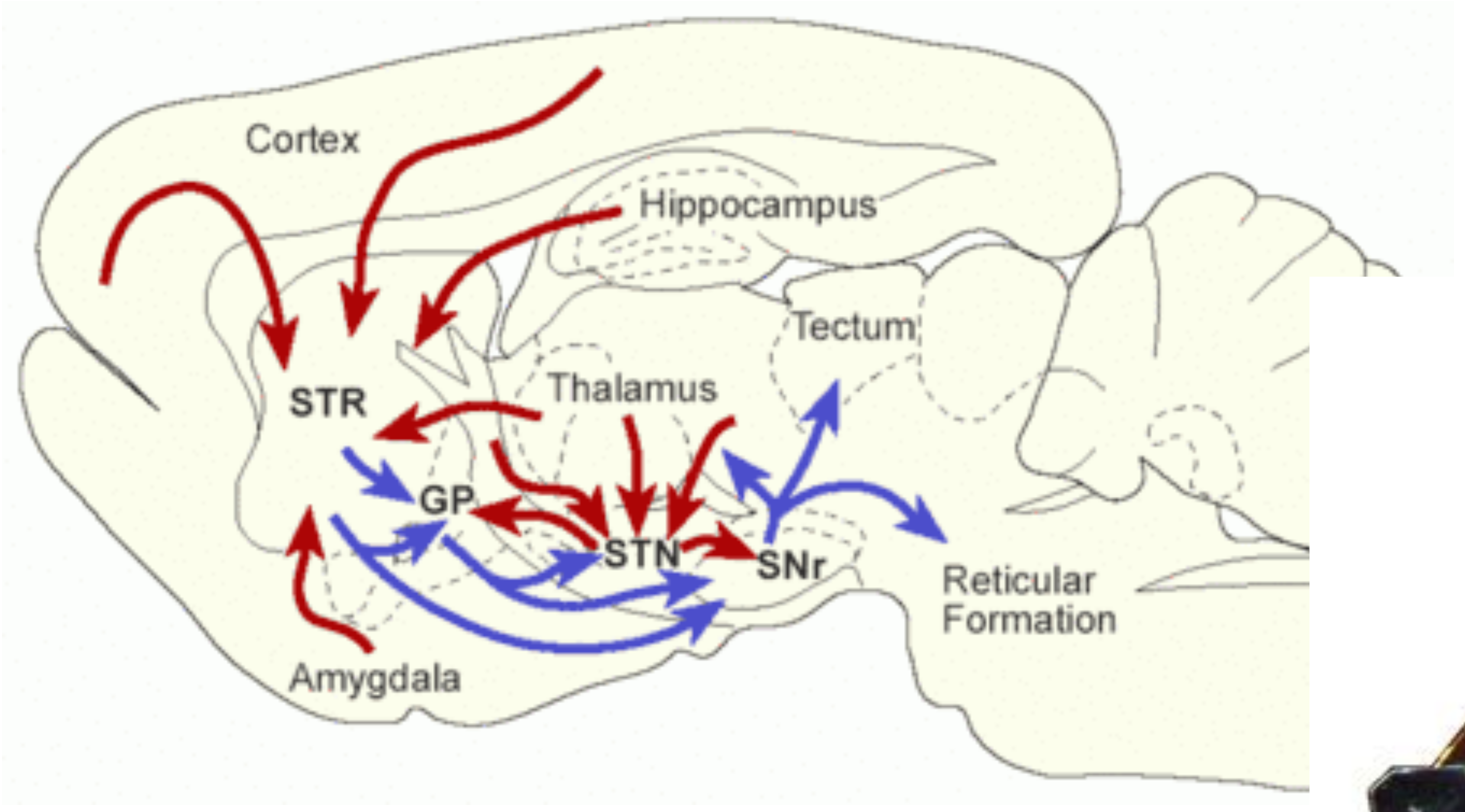


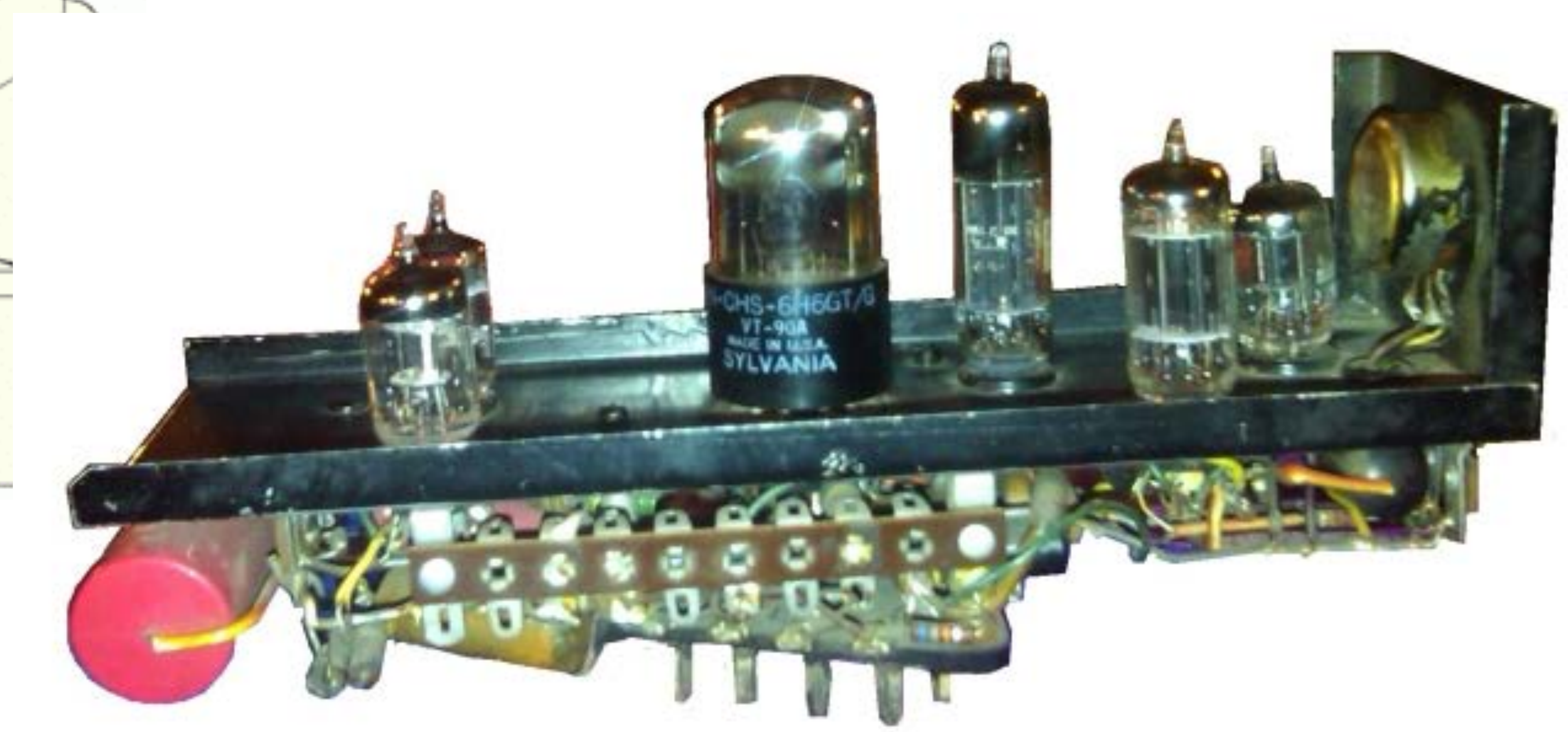
Two Main Recent Success Stories in AI

DL and RL come from neuroscience

R
L







Minsky's SNARC

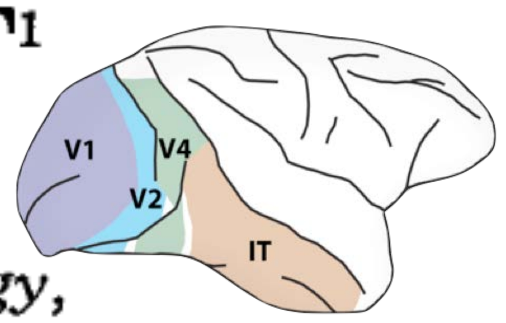
D
L

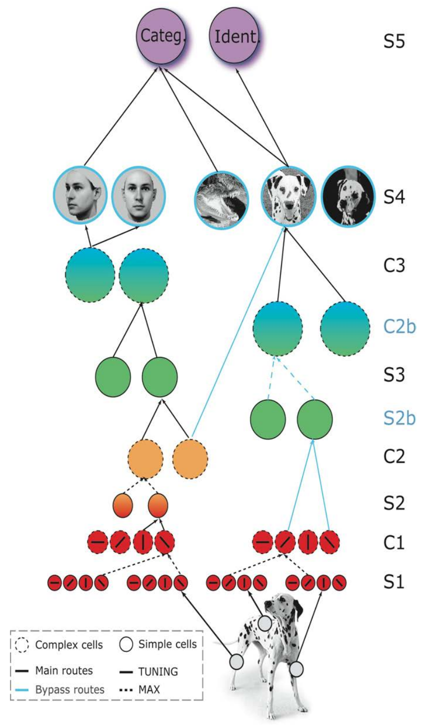
RECEPTIVE FIELDS AND FUNCTIONAL ARCHITECTURE IN TWO NONSTRIATE VISUAL AREAS (18 AND 19) OF THE CAT[1]

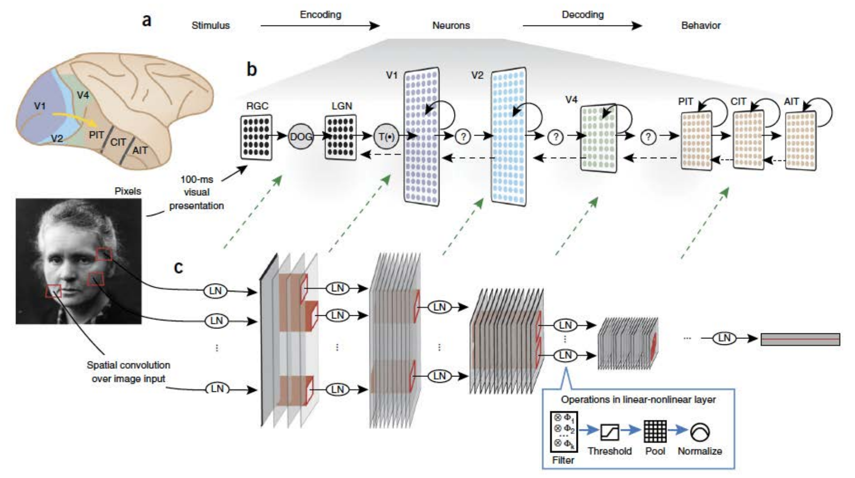
DAVID H. HUBEL AND TORSTEN N. WIESEL

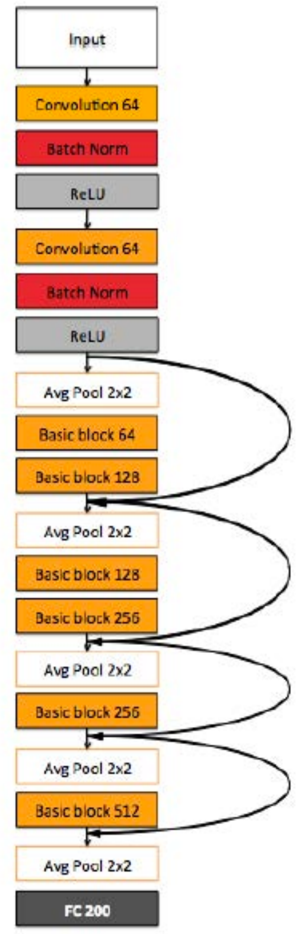
Neurophysiology Laboratory, Department of Pharmacology, Harvard Medical School, Boston, Massachusetts

(Received for publication August 24, 1964)









The Science of Intelligence

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

We need to make a basic effort leveraging
the old and new
science of intelligence:
neuroscience, cognitive science
and
combine it
with learning theory

INTERVIEW | SCIENCE | TECH

DeepMind's founder says to build better computer brains, we need to look at our own

What AI can learn from neuroscience, and neuroscience from AI

by James Vincent | @jjvincent | Jul 19, 2017, 12:00pm EDT

Illustration by James Bareham / The Verge

They point out that contemporary AI programs are extremely narrow in their abilities; that they're easily tricked, and simply don't possess those hard-to-define — but easy-to-spot skills we usually sum up as "common sense." They are, in short, not that intelligent.

The question is: how do we get to the next level? For Demis Hassabis, founder of Google's AI powerhouse DeepMind, the answer lies within us. Literally. In a review

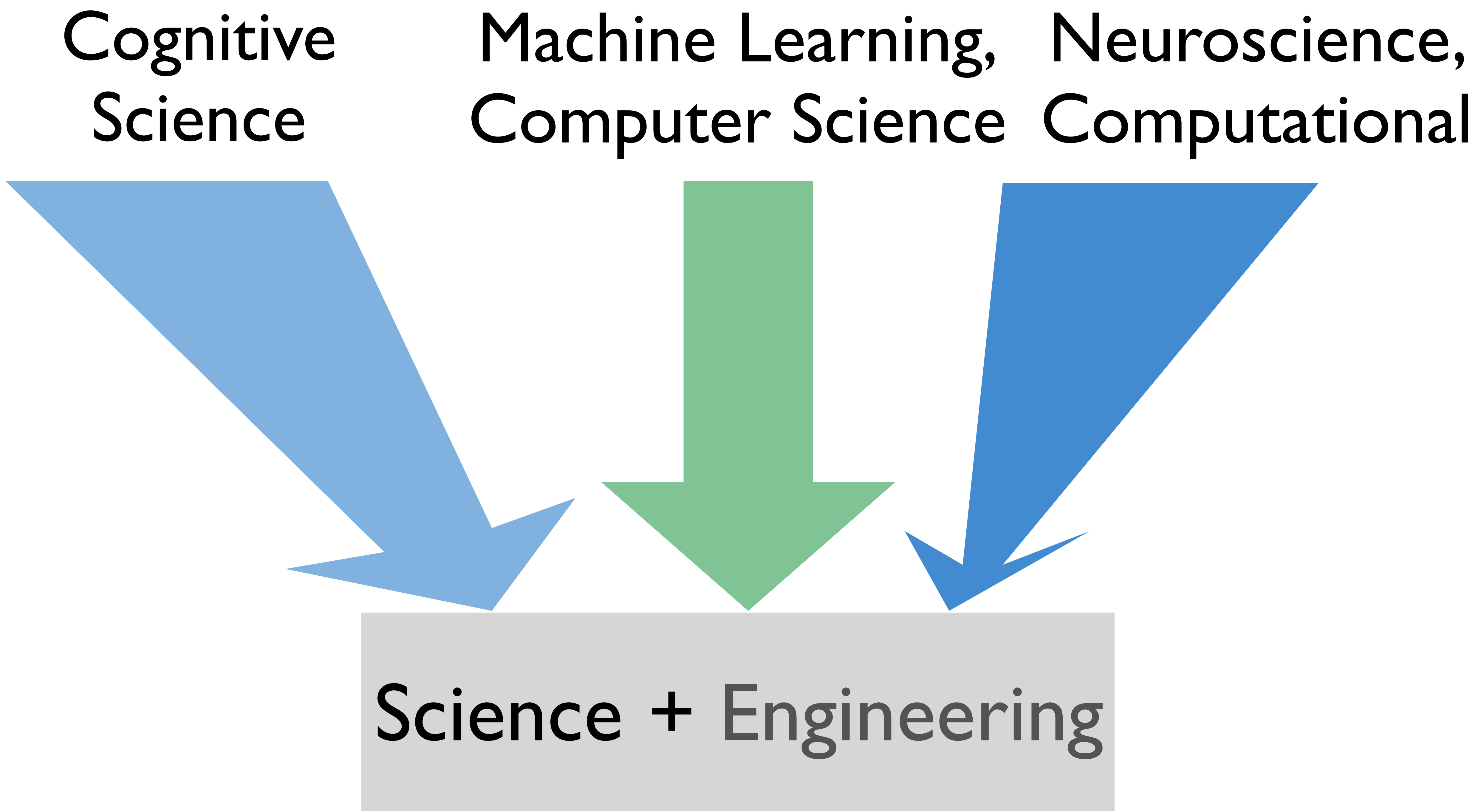
CENTER FOR Brains Minds+ Machines

CBMM: the Science and Engineering of Intelligence

CENTER FOR
Brains
Minds+
Machines

The Center for Brains, Minds and Machines (CBMM) is a multi-institutional NSF Science and Technology Center dedicated to the study of intelligence - how the brain produces intelligent behavior and how we may be able to replicate intelligence in machines.

Funding 2013-2023	**~$50M**
Research Institutions	**~4**
Educational Institutions	**12**
Faculty (CS+BCS+…)	**~23**
Researchers	**223**
Publications	**397**



Research, Education & Diversity Partners

MIT
Boyden, Desimone, DiCarlo, Kanwisher, Katz, McDermott, Poggio, Rosasco, Sassanfar, Saxe, Schulz, Tegmark, Tenenbaum, Ullman, Wilson, Winston, Torralba

Harvard
Blum, Gershman, Kreiman, Livingstone, Nakayama, Sompolinsky, Spelke

Boston Children's Hospital
Kreiman

Florida International U.
Finlayson

Harvard Medical School
Kreiman, Livingstone

Howard U.
Chouika, Manaye, Rwebangira, Salmani

Hunter College
Chodorow, Epstein, Sakas, Zeigler

Johns Hopkins U.
Yuille

Queens College
Brumberg

Rockefeller U.
Freiwald

Stanford U.
Goodman

Universidad Central Del Caribe (UCC)
Jorquera

University of Central Florida
McNair Program

UMass Boston
Blaser, Ciaramitaro, Pomplun, Shukla

UPR - Mayagüez
Santiago, Vega-Riveros

UPR – Río Piedras
Garcia-Arraras, Maldonado-Vlaar, Megret, Ordóñez, Ortiz-Zuazaga

Wellesley College
Hildreth, Wiest, Wilmer

International and Corporate Partners

A*STAR
Chuan Poh Lim

Genoa U.
Verri, Rosasco

Hebrew U.
Weiss

Kaist
Sangwan Lee

IIT
Cingolani

MPI
Bülthoff

Weizmann
Ullman

Google

IBM

Microsoft

Orcam

Siemens

Honda

Fujitsu

NVIDIA

DeepMind

Boston Dynamics

GE

Schlumberger

Mobileye

Intel

EAC Meeting: March 19, 2019

Demis Hassabis, *DeepMind*

Charles Isbell, Jr., *Georgia Tech*

Christof Koch, *Allen Institute*

Fei-Fei Li, *Stanford*

Lore McGovern, *MIBR, MIT*

Joel Oppenheim, *NYU*

Pietro Perona, *Caltech*

Marc Raibert, *Boston Dynamics*

Judith Richter, *Medinol*

Kobi Richter, *Medinol*

Dan Rockmore, *Dartmouth*

Amnon Shashua, *Mobileye*

David Siegel, *Two Sigma*

Susan Whitehead, *MIT Corporation*

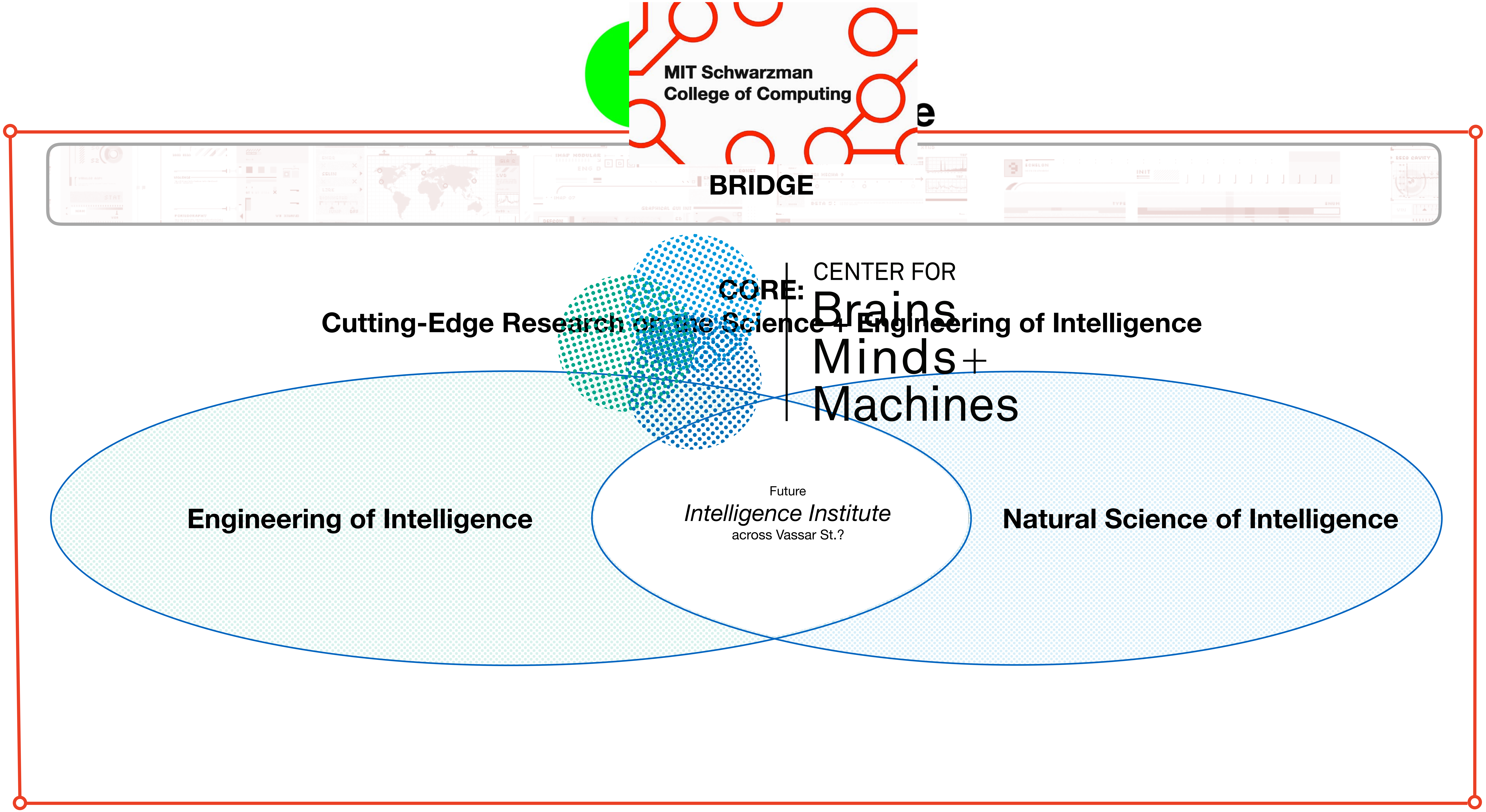
Jim Pallotta, *The Raptor group*

Summer Course at Woods Hole: Our flagship initiative

Brains, Minds & Machines Summer Course

Gabriel Kreiman + Boris Katz

A community of scholars is being formed:


MIT Schwarzman
College of Computing
BRIDGE
CORE:
Cutting-Edge Research on the Science + Engineering of Intelligence
CENTER FOR
Brains
Minds+
Machines
Engineering of Intelligence
Future
Intelligence Institute
across Vassar St.?
Natural Science of Intelligence

Summary

- Motivations for this course: a golden age for new AI, the key role of Machine Learning, CBMM

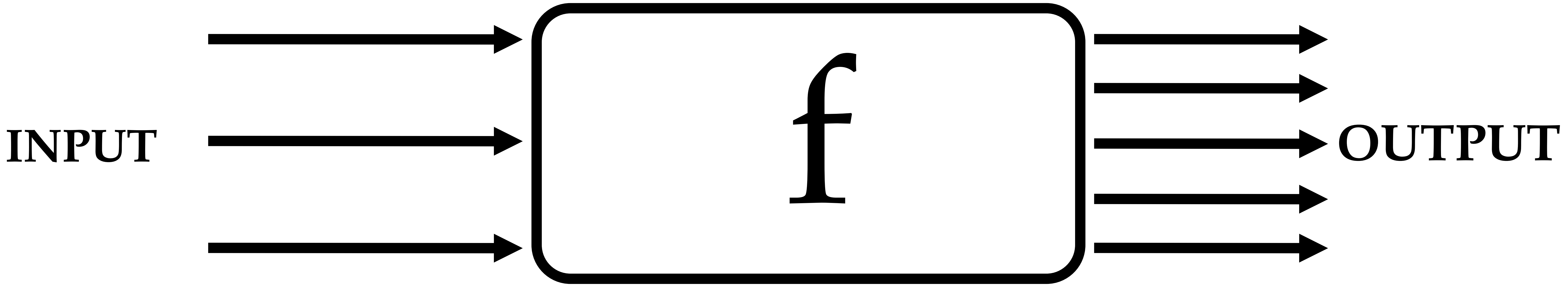
Summary: I told you about the present great success of ML, its connections with neuroscience, its limitations for full AI. I then told you that we need to connect to neuroscience if we want to realize <u>real</u> AI, in addition to understanding our brain. BTW, even without this extension, the next few years will be a golden age for ML applications.

Today's overview

- Course description/logistic
- Motivations for this course: a golden age for new AI, the key role of Machine Learning, CBMM, the MIT Quest: ***Intelligence, the Grand Vision***
- A bit of history: Statistical Learning Theory and Applications
- Deep Learning

Statistical Learning Theory

Statistical Learning Theory: **supervised** learning (~1980-today)



Given a set of l examples (data)

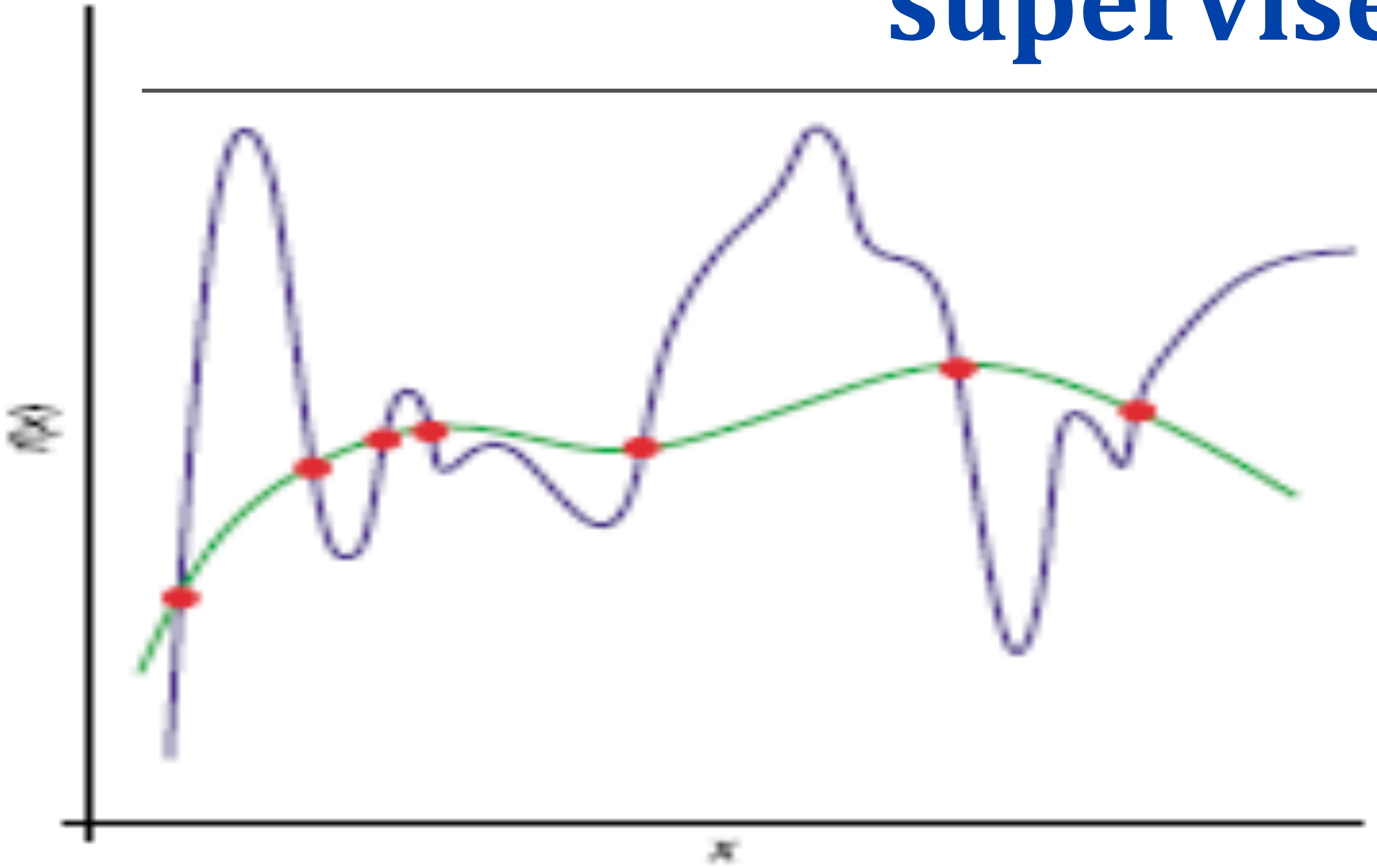
$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$$

Question: find function f such that

$$f(x) = \hat{y}$$

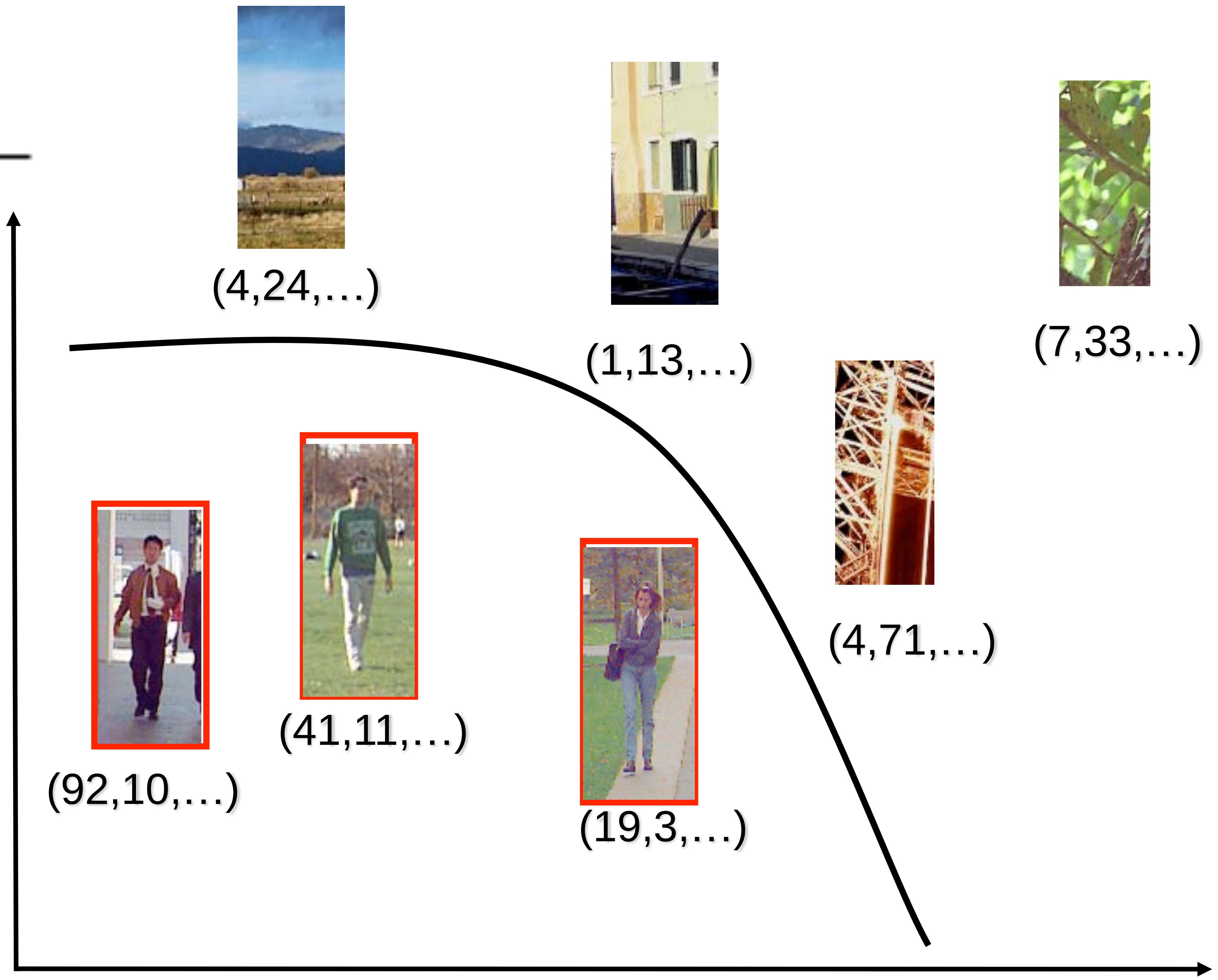
is a **good predictor** of y for a **future** input x (fitting the data is **not** enough!)

Statistical Learning Theory: supervised learning



Regression

Classification

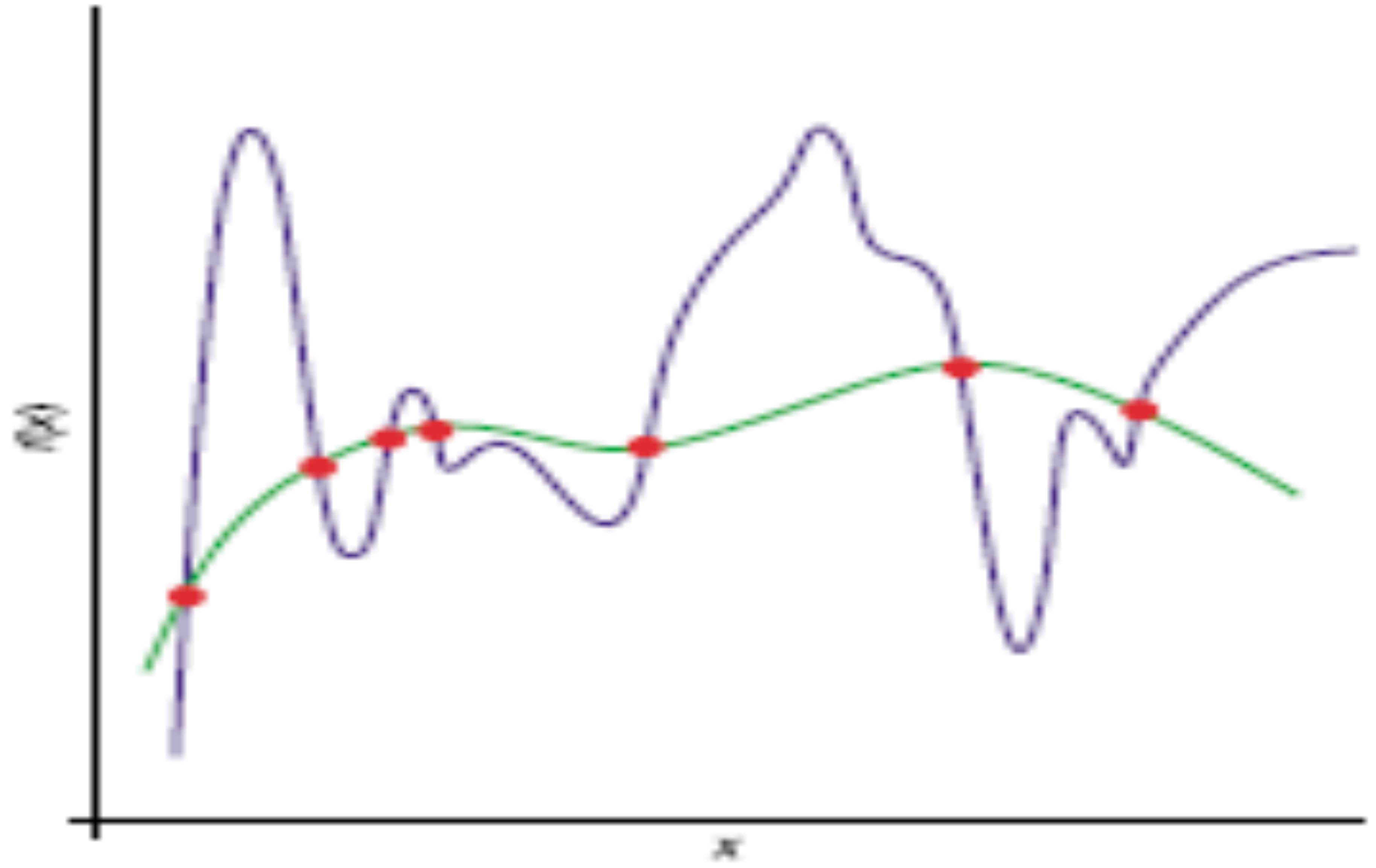


Statistical Learning Theory: prediction, not description

= data from f

= function f

= approximation of f



Intuition: Learning from data to predict well the value of the function where there are no data

Statistical Learning Theory: supervised learning

There is an unknown **probability distribution** on the product space $Z = X \times Y$, written $\mu(z) = \mu(x, y)$. We assume that X is a compact domain in Euclidean space and Y a bounded subset of $\mathbb{R}$. The **training set** $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} = \{z_1, ...z_n\}$

consists of n samples drawn i.i.d. from μ.

$\mathcal{H}$ is the **hypothesis space**, a space of functions $f : X \to Y$.

A **learning algorithm** is a map $L : Z^n \to \mathcal{H}$ that looks at S and selects from $\mathcal{H}$ a function $f_S : \mathbf{x} \to y$ such that $f_S(\mathbf{x}) \approx y$ *in a predictive way*.

Statistical Learning Theory

Given a function f, a loss function V, and a probability distribution μ over Z, the **expected or true error** of f is:

$$I[f] = \mathbb{E}_Z V[f, z] = \int_Z V(f, z) d\mu(z) \quad (1)$$

which is the **expected loss** on a new example drawn at random from μ.

The **empirical error** of f is:

$$I_S[f] = \frac{1}{n} \sum V(f, z_i) \quad (2)$$

A very natural requirement for f_S is distribution independent **generalization**

$$\forall \mu, \lim_{n \to \infty} |I_S[f_S] - I[f_S]| = 0 \textit{ in probability} \quad (3)$$

In other words, the training error for the solution must converge to the expected error and thus be a "proxy" for it.

Statistical Learning Theory: foundational theorems

Conditions for <u>generalization</u> and <u>well-posedness/stability</u>
in learning theory
have deep, almost philosophical, implications:

they can be regarded as <u>equivalent conditions</u> that guarantee a theory to be predictive and scientific

- theory must be chosen from a small hypothesis set (~ Occam razor, VC dimension,…)
- theory should not change much with new data...most of the time (stability)

One of the key msgs of the 80'-90' from learning theory: do not overfit the data because you will not predict well!
Models must be constrained, their capacity controlled!
Astronomy, not astrology!

Classical algorithm:
Regularization in RKHS (eg. kernel machines)

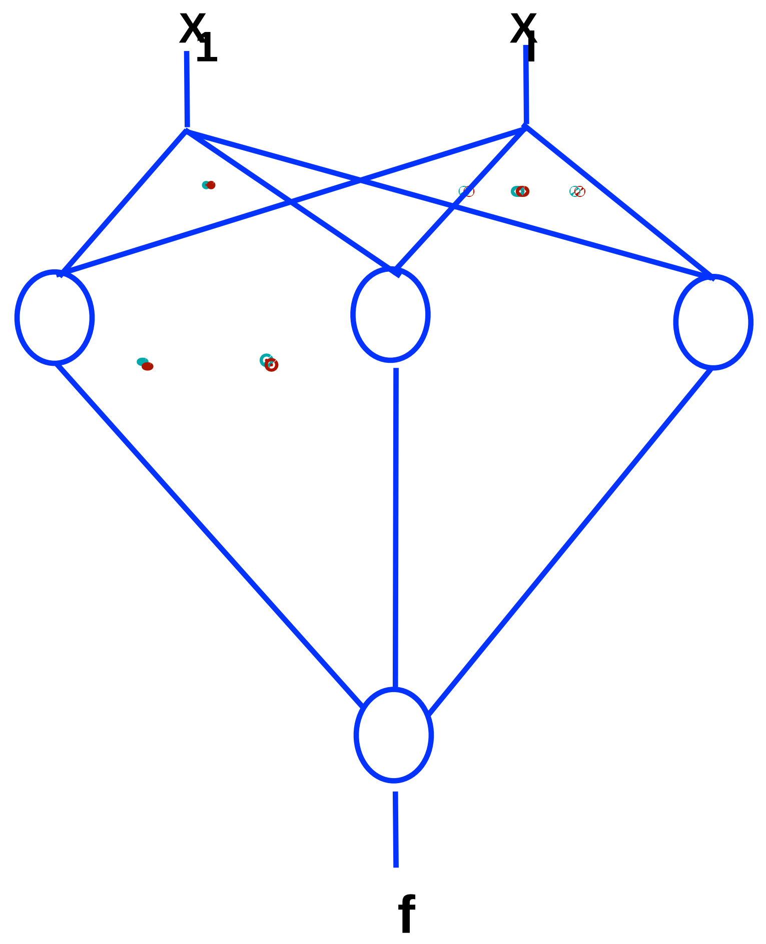
$$\min_{f \in H} \left[\frac{1}{n} \sum_{i=1}^{n} V(f(x_i) - y_i) + \lambda \|f\|_K^2 \right]$$

The regularization term controls the complexity of the function in terms of its RKHS norm

implies

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

Classical kernel machines — such as SVMs — correspond to <u>shallow</u> networks



Summary

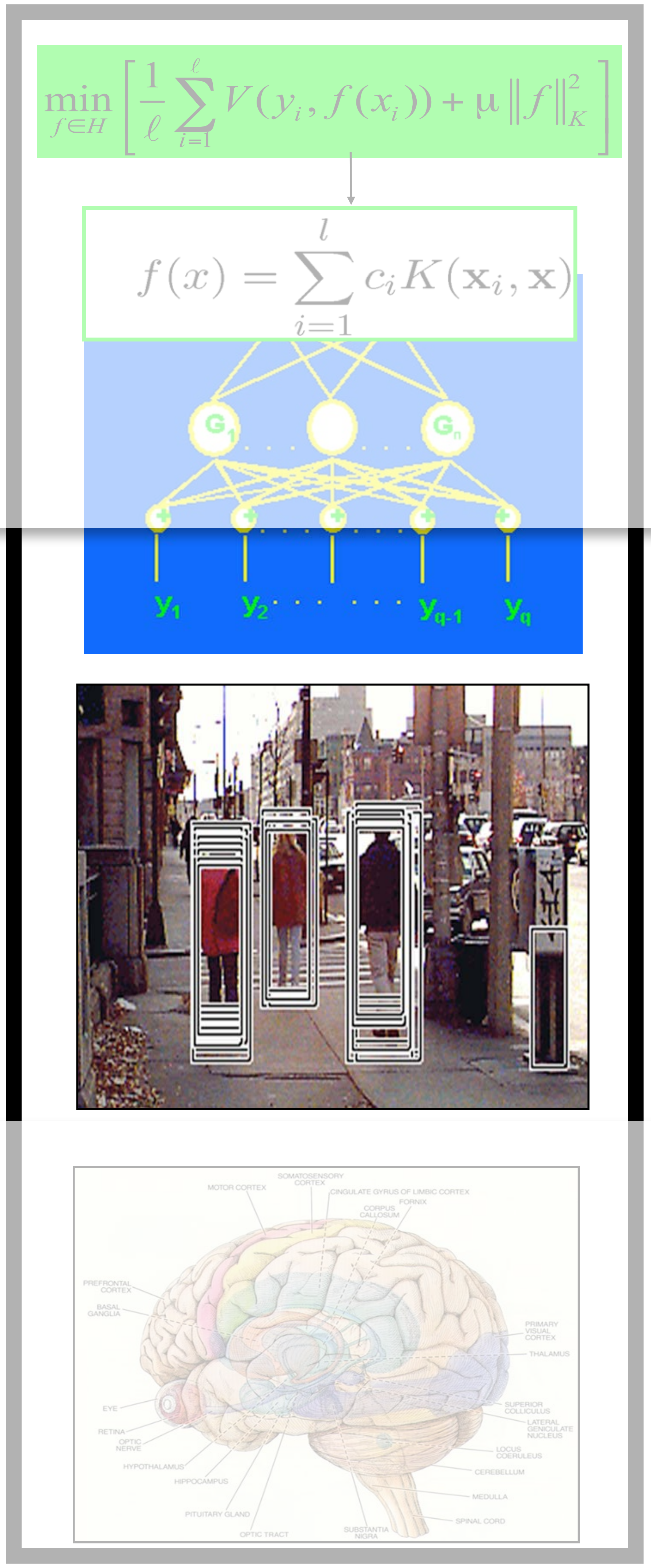
Bits of history: Statistical Learning Theory

Summary: I told you about learning theory and predictivity. I told you about kernel machines and shallow networks.

Historical perspective:
Examples of old Applications

Kah-Kay Sung
around ~1990

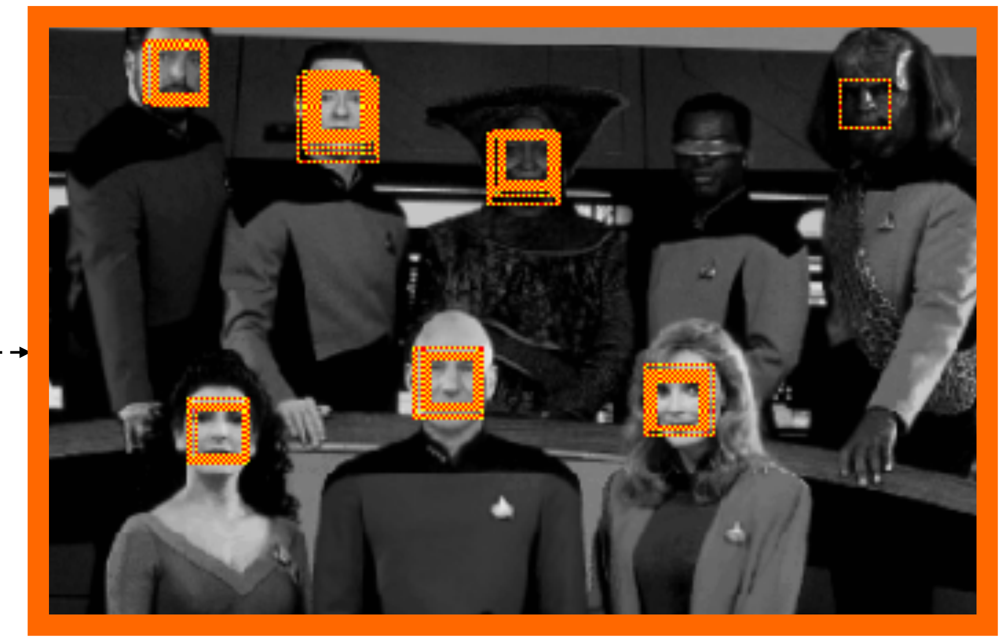
Engineering of Learning



LEARNING THEORY
+
ALGORITHMS

Theorems on foundations of learning

Predictive algorithms



Face detection has been available in digital cameras for a few years now

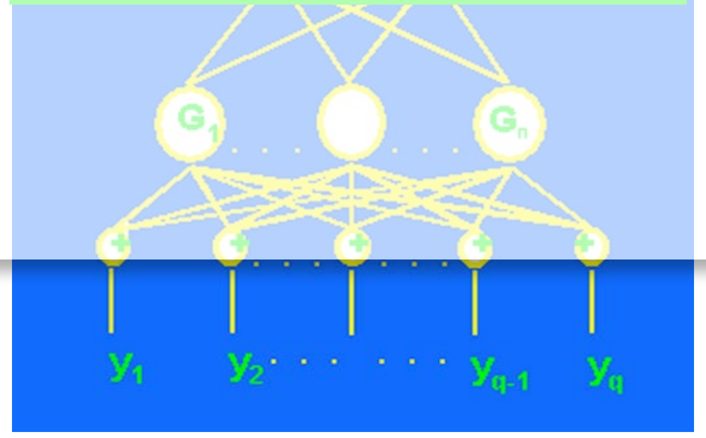
COMPUTATIONAL NEUROSCIENCE: models+experiments

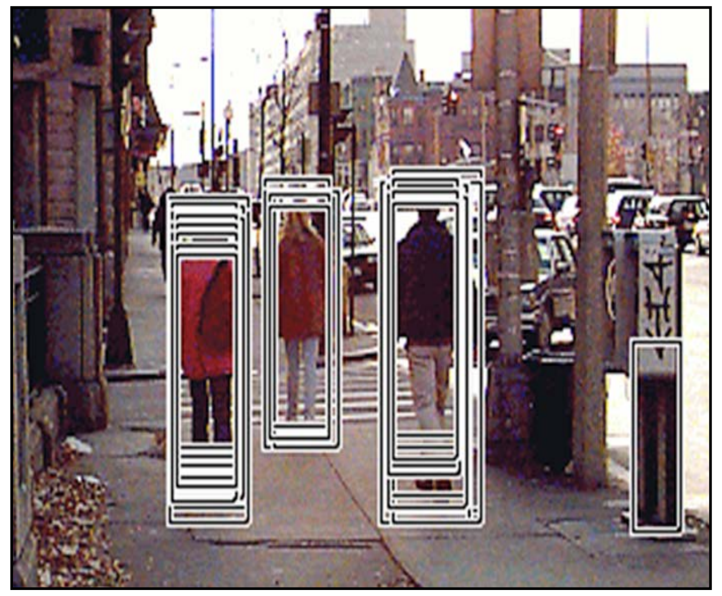
How visual cortex works

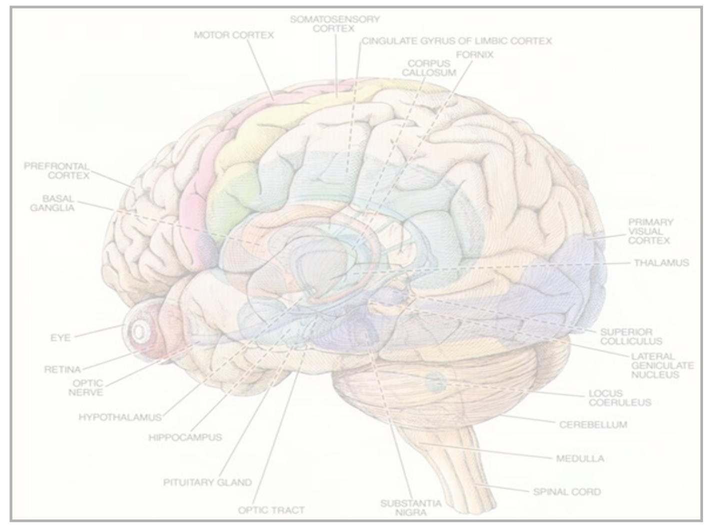
Engineering of Learning

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

$$f(x) = \sum_{i=1}^{l} c_i K(\mathbf{x}_i, \mathbf{x})$$







LEARNING THEORY
+
ALGORITHMS

Theorems on foundations of learning

Predictive algorithms

around ~1997

Pedestrian detection

Papageorgiou&Poggio, 1997, 2000
also Kanade&Scheiderman

COMPUTATIONAL NEUROSCIENCE: models+experiments

How visual cortex works

2015

~1995

Some other examples of past ML applications from my lab (from 1990 to ~2001)

Computer Vision

- Face detection
- Pedestrian detection
- Scene understanding
- Video categorization
- Video compression
- Pose estimation

Graphics

Speech recognition

Speech synthesis

Decoding the Neural Code

Bioinformatics

Text Classification

Artificial Markets

Stock option pricing

….

Learning: bioinformatics

New feature selection SVM:

Only 38 training examples, 7100 features

AML vs ALL: 40 genes 34/34 correct, 0 rejects.
5 genes 31/31 correct, 3 rejects of which 1 is an error.

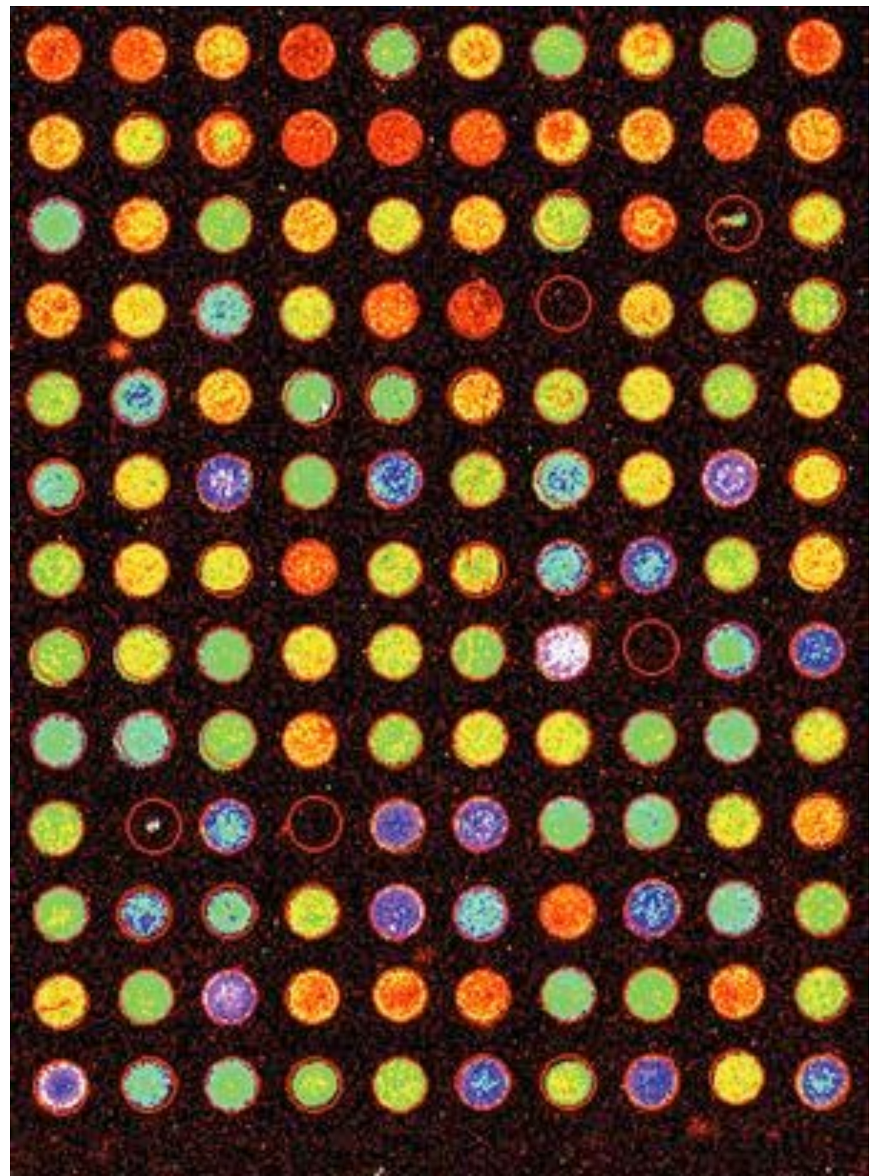
around ~2000

A.I. Memo No.1677
C.B.C.L Paper No.182

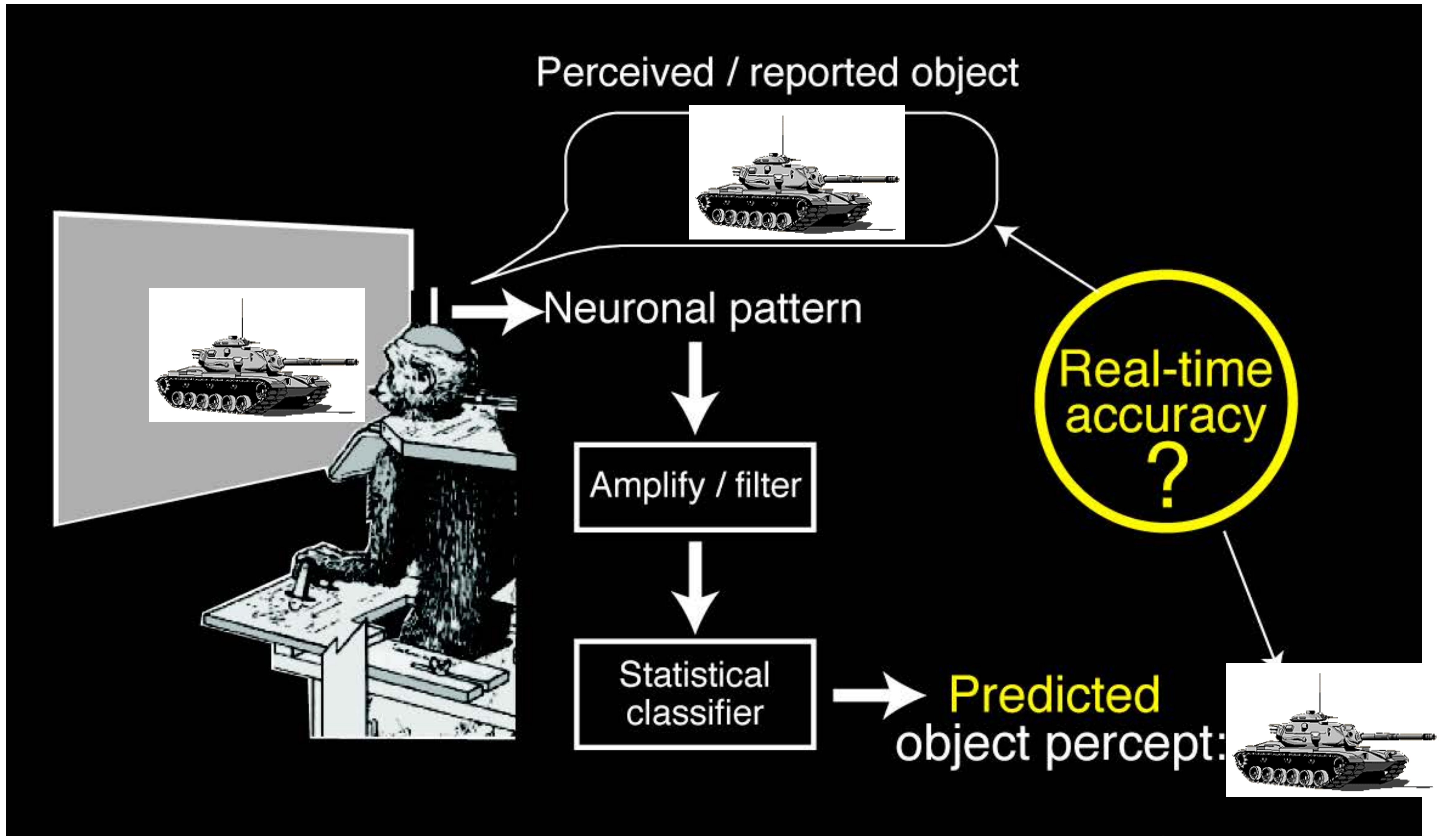
Support Vector Machine Classification of Microarray Data

S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J.P. Mesirov, and T. Poggio

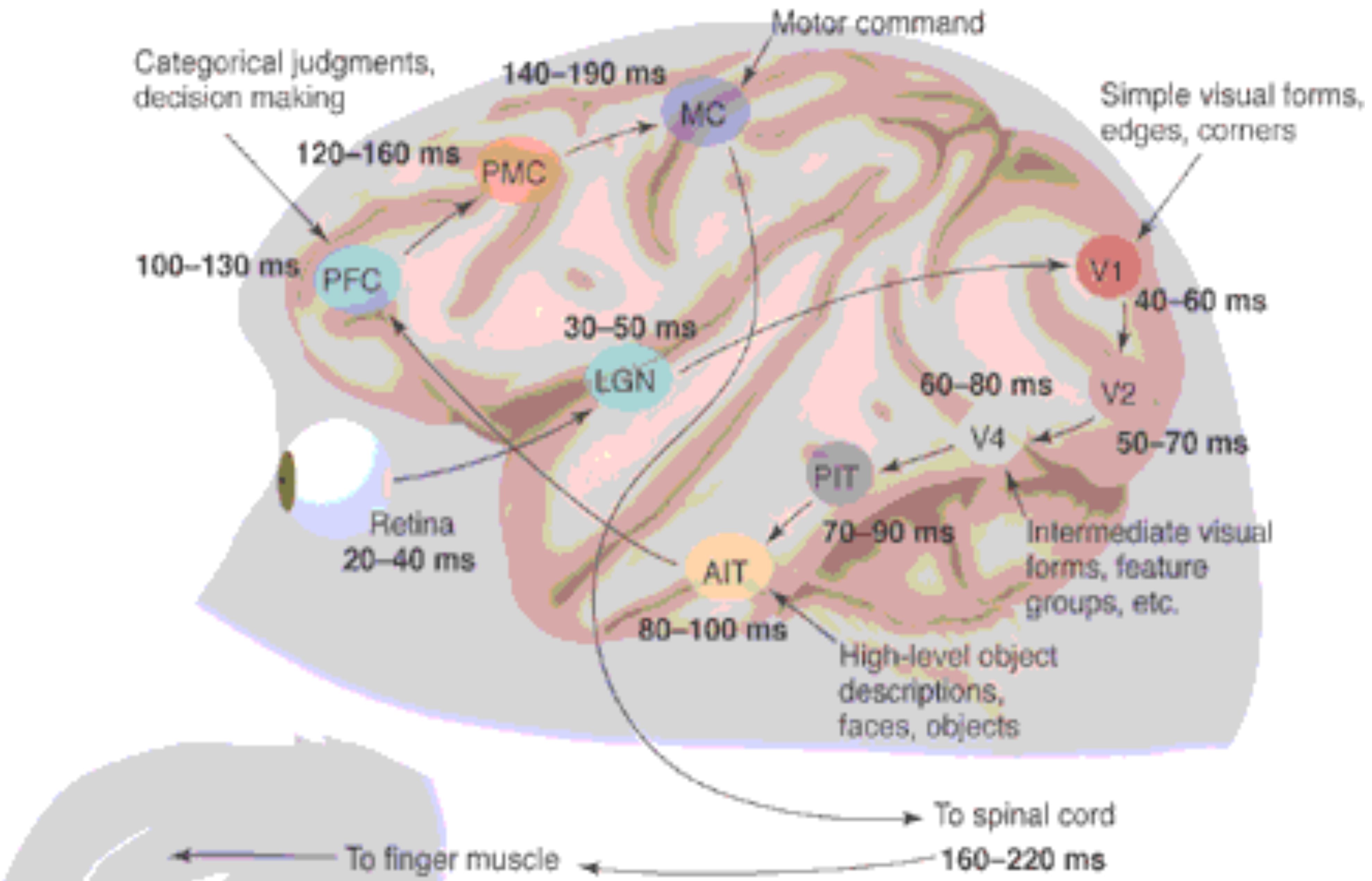
Pomeroy, S.L., P. Tamayo, M. Gaasenbeek, L.M. Sturia, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, M.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander and T.R. Golub. Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression, *Nature*, 2002.



Decoding the neural code: Matrix-like read-out from the brain

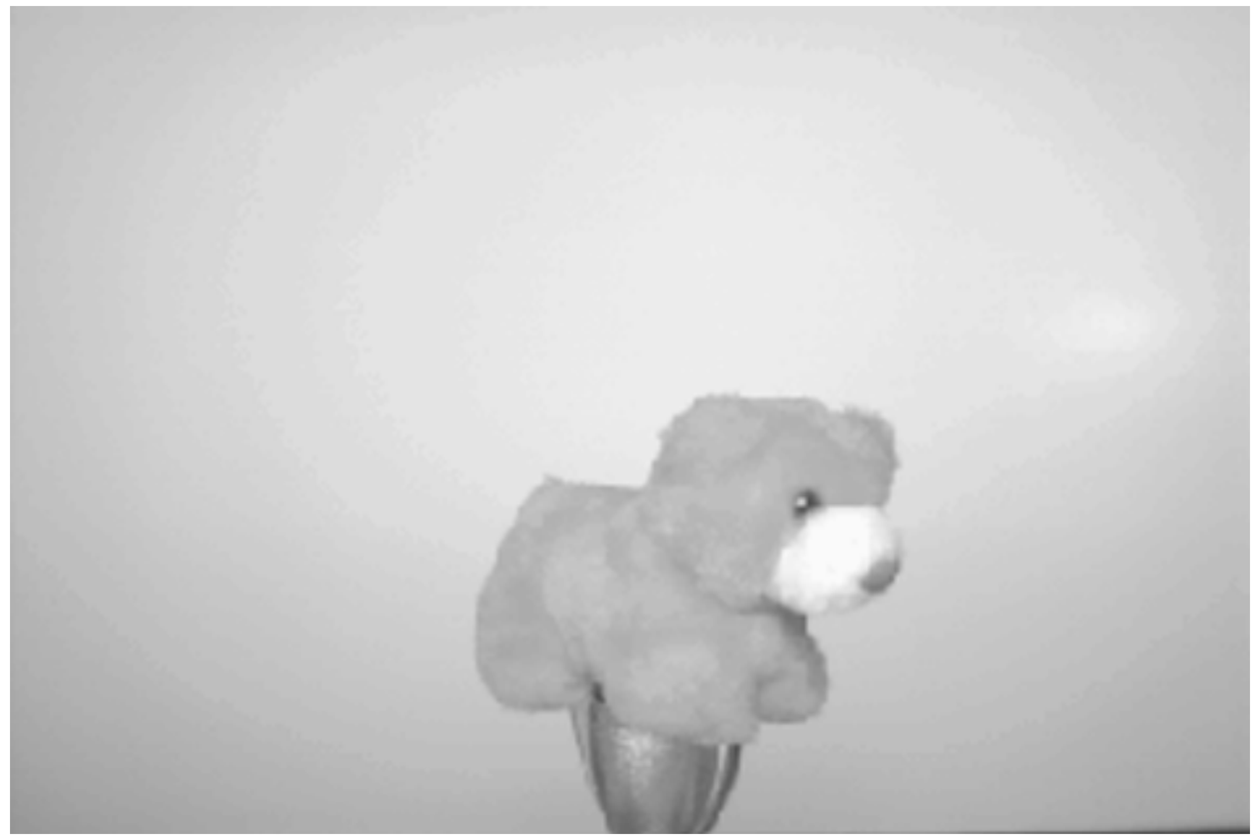


Science around ~2005



Learning: image analysis

⇒ **Bear (0° view)**



⇒ **Bear (45° view)**

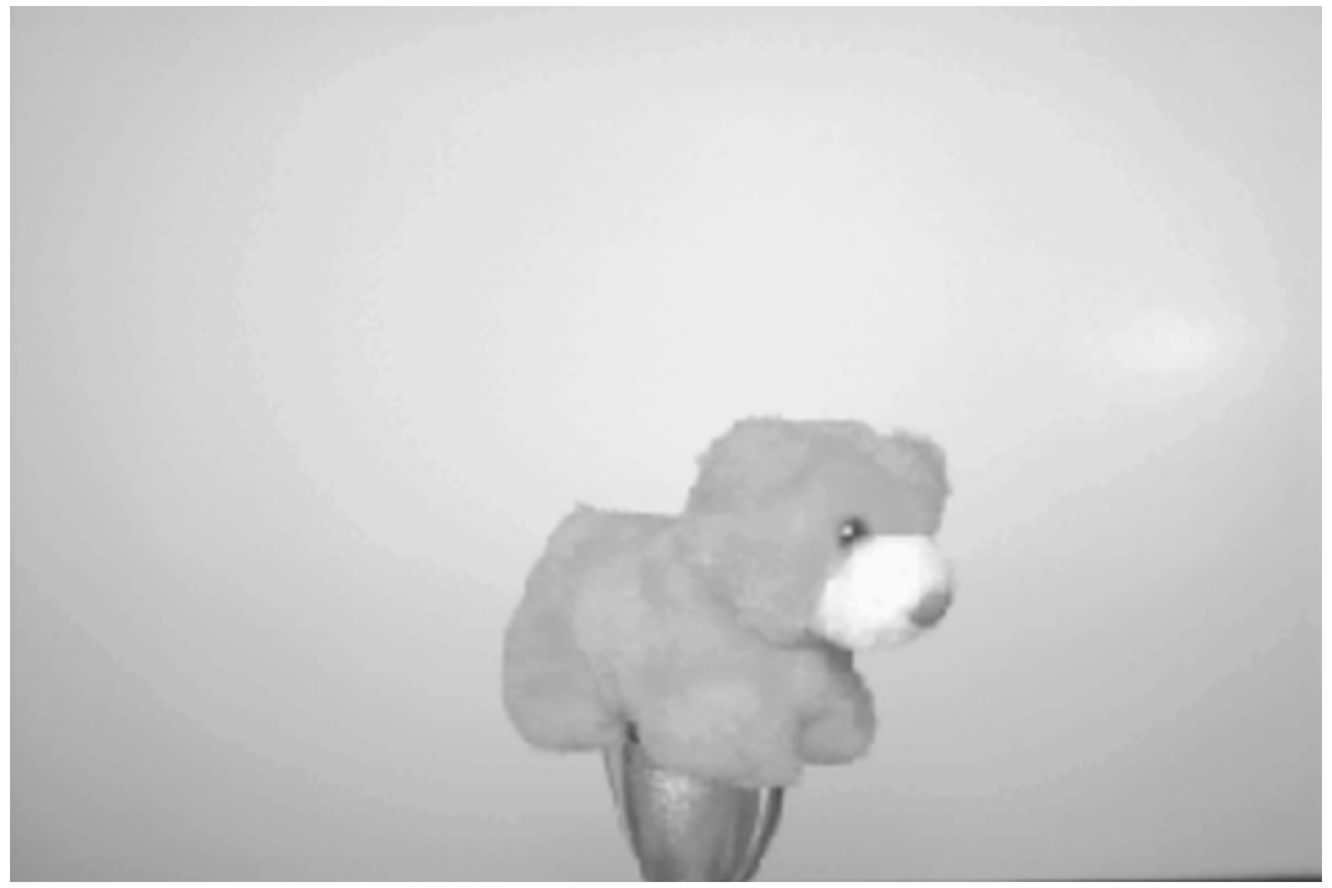
around ~1995

Learning: image synthesis

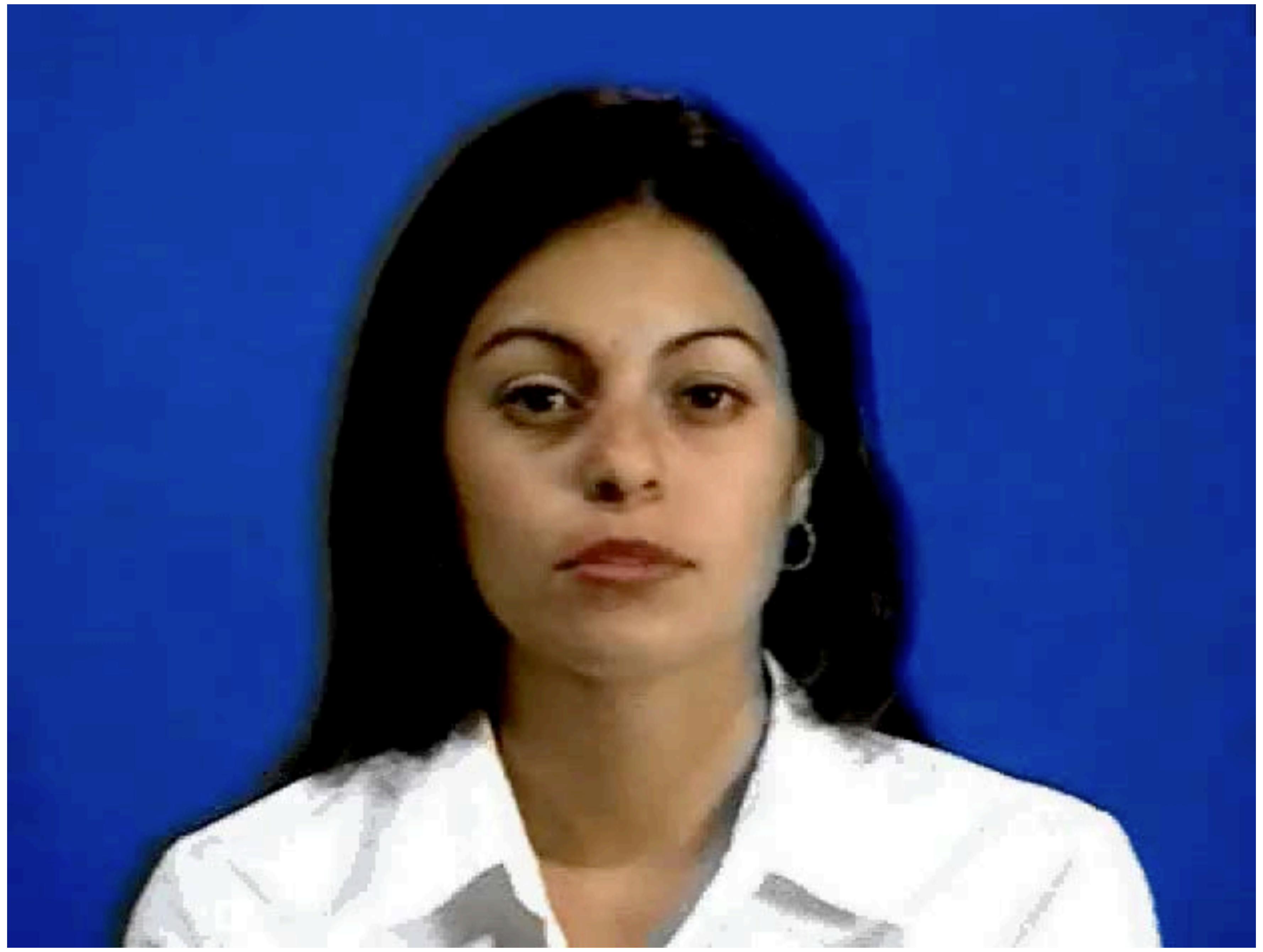
UNCONVENTIONAL GRAPHICS

Θ = 0° view $\Rightarrow$

Θ = 45° view $\Rightarrow$



Extending the same basic learning techniques (in 2D): Trainable Videorealistic Face Animation



Mary101

A- more in a moment

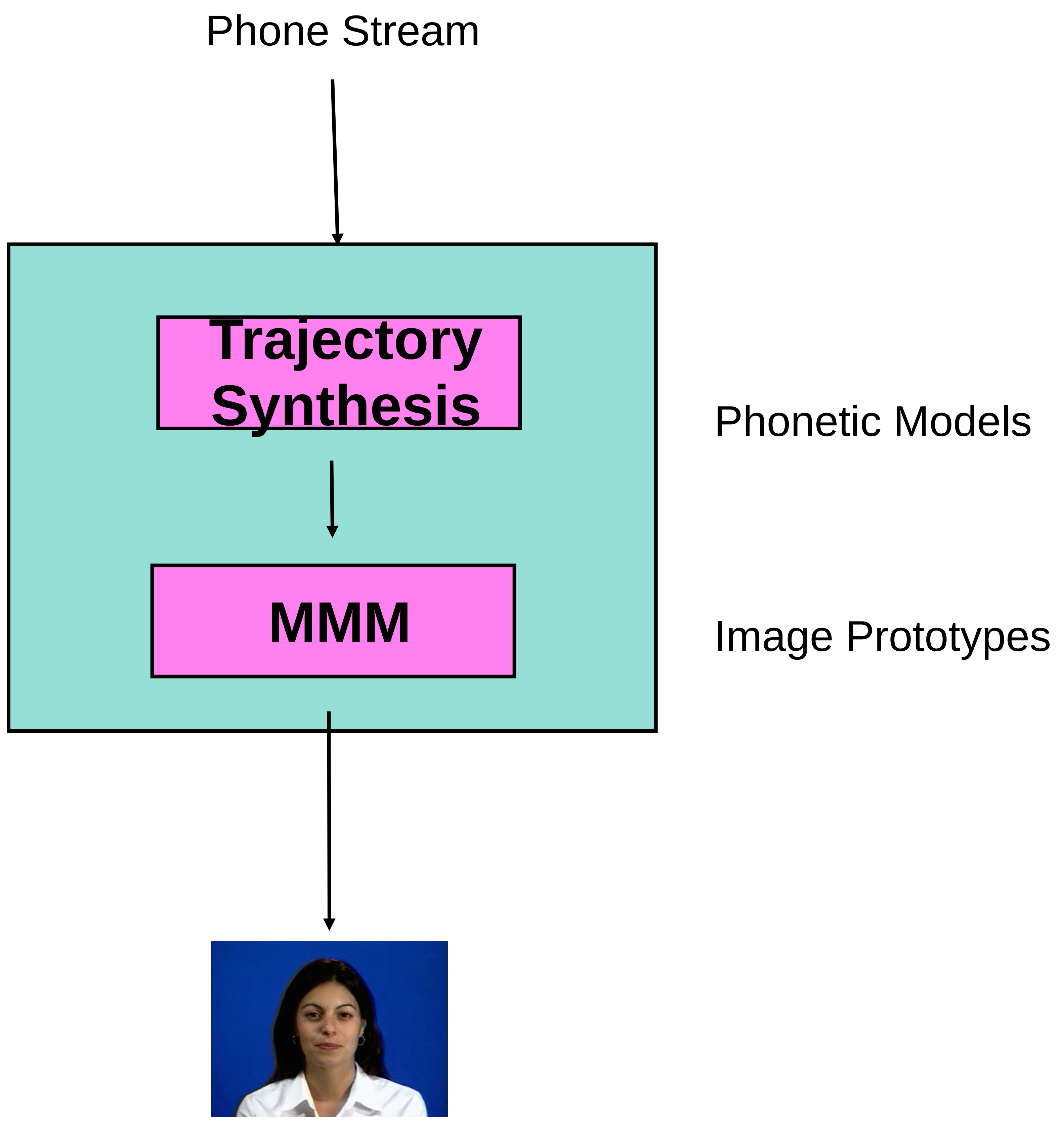
Tony Ezzat,Geiger, Poggio, **SigGraph 2002**

1. Learning

System learns from 4 mins of video face appearance (Morphable Model) and speech dynamics of the person

2. Run Time

For any speech input the system provides as output a synthetic video stream



B-Dido

C-Hikaru

D-Denglijun

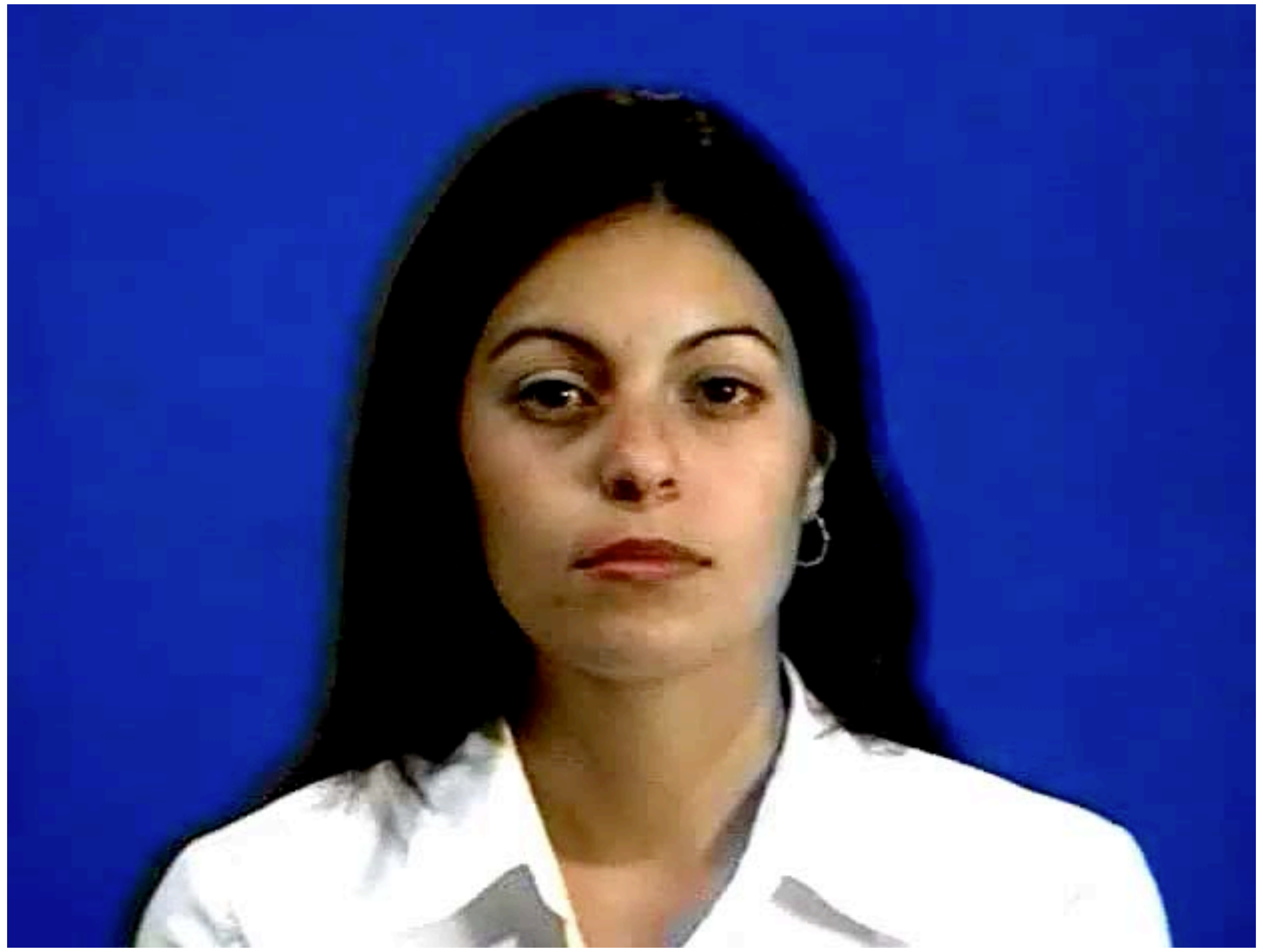
E-Marylin

G-Katie

H-Rehema

I-Rehemax

A Turing test: what is real and what is synthetic?



L-real-synth

A Turing test: what is real and what is synthetic?

Experiment	# subjects	% correct	t	p<
Single pres.	22	54.3%	1.243	0.3
Fast single pres.	21	52.1%	0.619	0.5
Double pres.	22	46.6%	-0.75	0.5

Table 1: Levels of correct identification of real and synthetic sequences. t represents the value from a standard t-test with significance level of p<.

Tony Ezzat,Geiger, Poggio, **SigGraph 2002**

Similar to today's GANs

Summary

- Bits of history: old applications

Summary: I told you about old applications of ML, mainly kernel machines to give a feeling for how broadly powerful is the supervised learning approach: you can apply it to visual recognition, to decode neural data, to medical diagnosis, to finance, even to graphics. I also wanted to make you aware that ML does not start with deep learning and certainly does not finish with it.

Today’s overview

- Course description/logistic

- Motivations for this course: a golden age for new AI, the key role of Machine Learning, CBMM, the MIT Quest: ***Intelligence, the Grand Vision***

- Bits of history: Statistical Learning Theory and Applications

- Deep Learning bits

Deep Learning

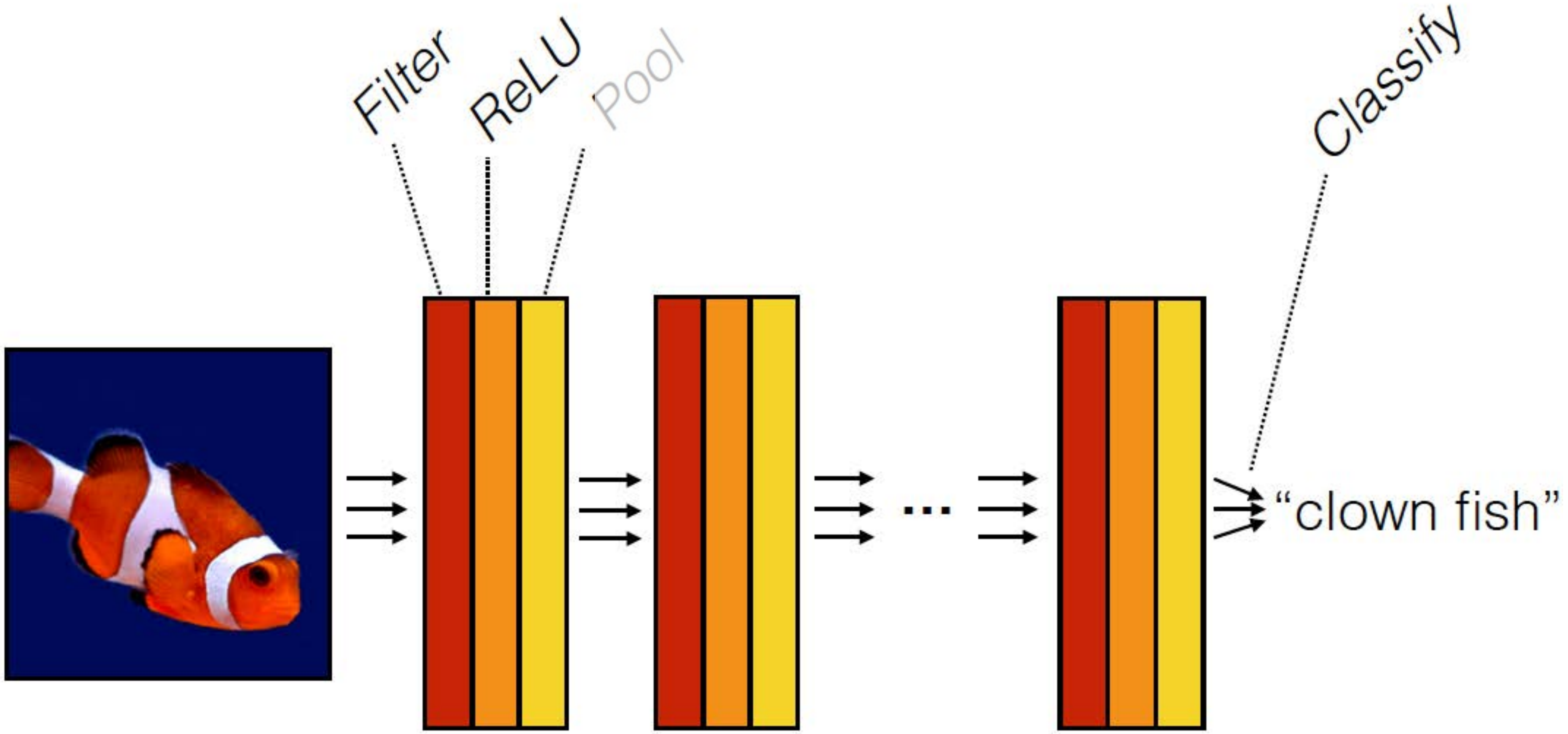
9.520/6.860

- classical regularization (regularized least squares, SVM, logistic regression, square and exponential loss), stochastic gradient methods, implicit regularization and minimum norm solutions. Regularization techniques, Kernel machines, batch and online supervised learning, sparsity.

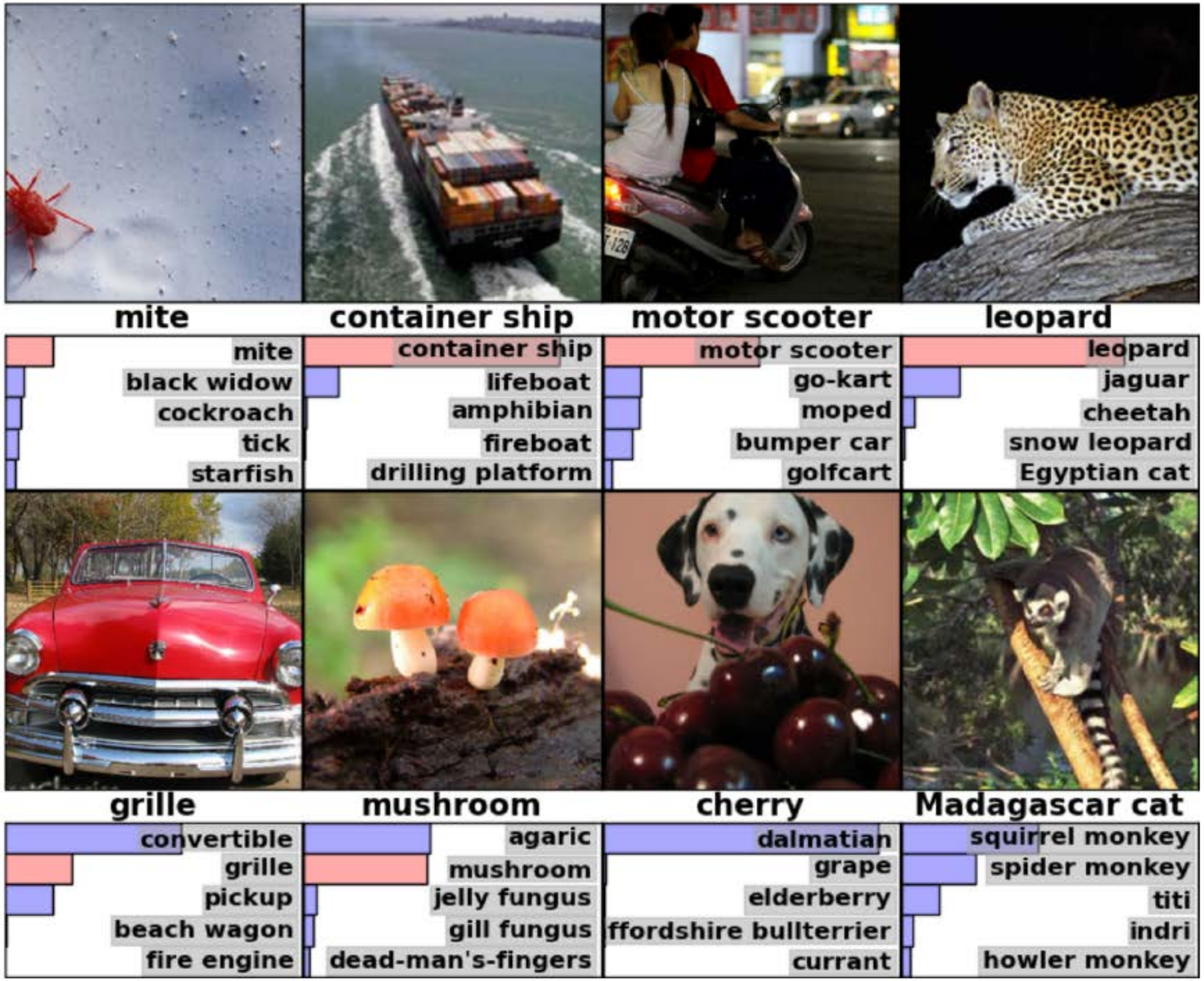
- Classical concepts like generalization, uniform convergence and Rademacher complexities will be developed, together with topics such as surrogate loss functions for classification, bounds based on margin,, and pstabilityrivacy.

- Theoretical frameworks addressing three key puzzles in deep learning: approximation theory -- which functions can be represented more efficiently by deep networks than shallow networks-- optimization theory -- why can stochastic gradient descent easily find global minima -- and machine learning -- how generalization in deep networks used for classification can be explained in terms of complexity control implicit in gradient descent. It will also discusses connections with the architecture of the brain, which was the original inspiration of the layered local connectivity of modern networks and may provide ideas for future developments and revolutions in networks for learning.

Computation in a neural net



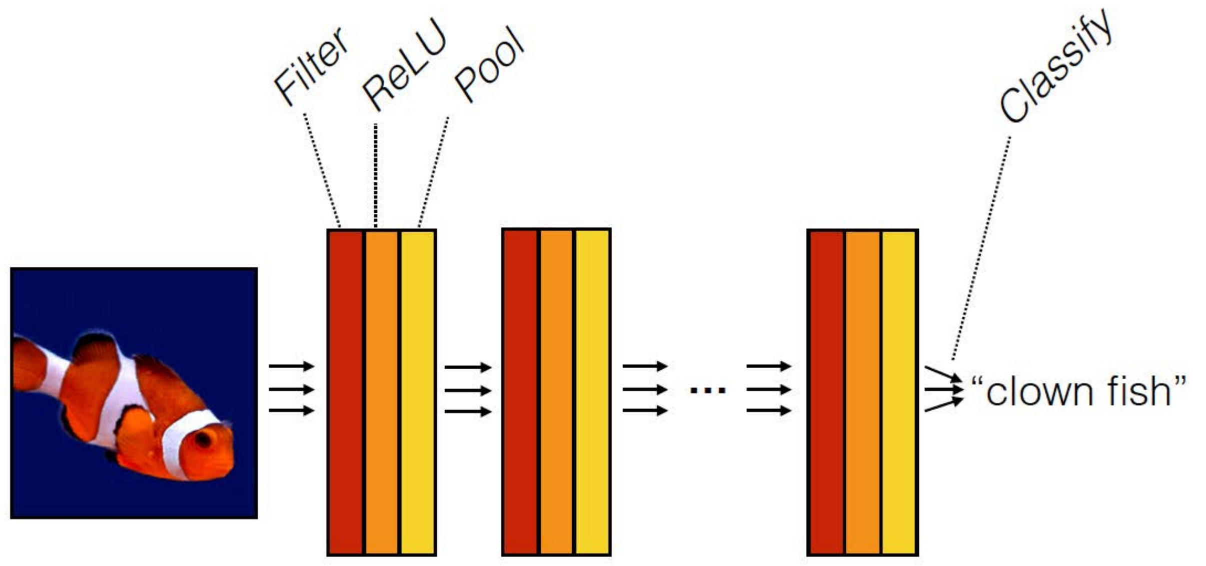
$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$



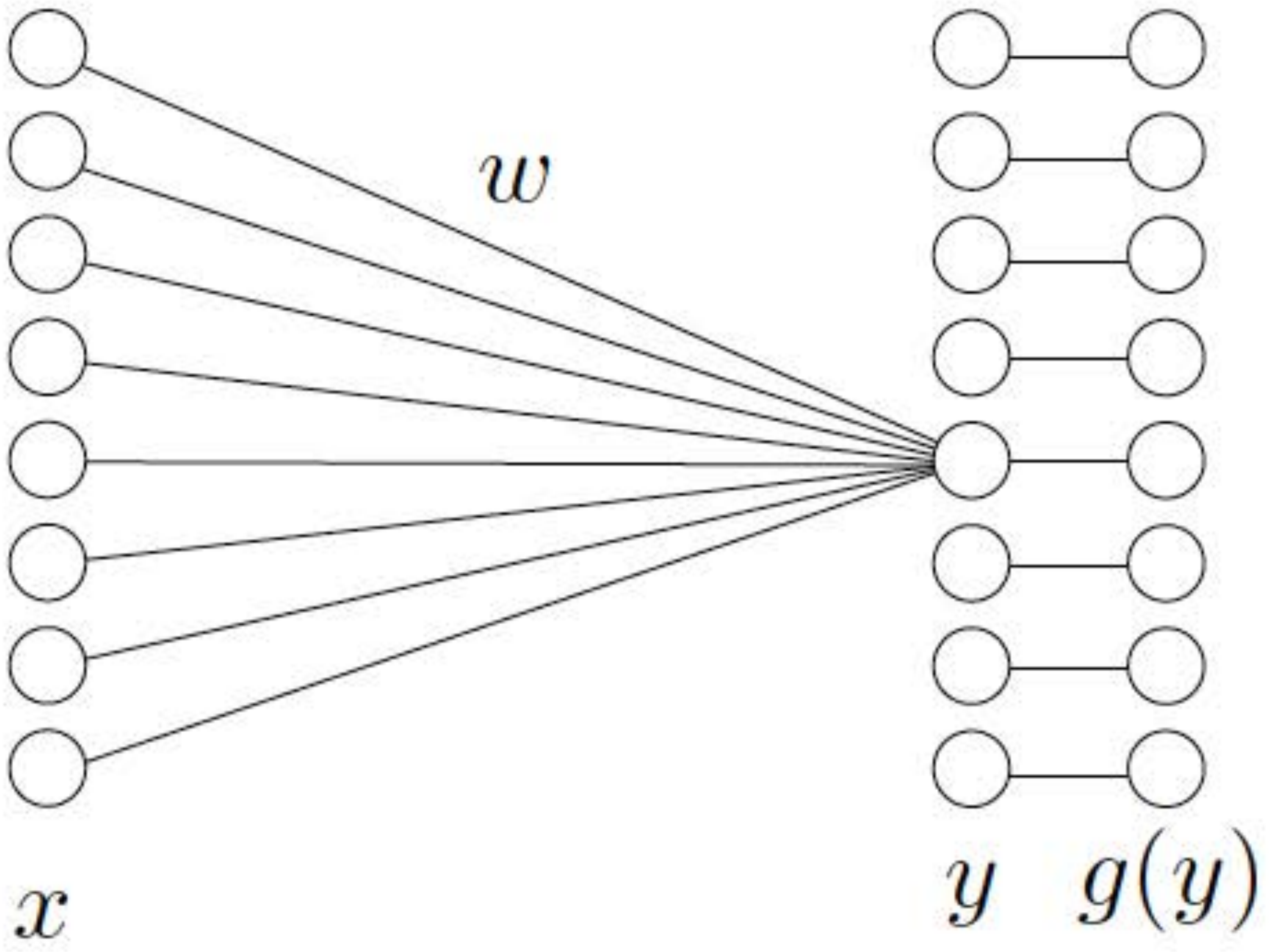
Krizhevsky et al. NIPS 2012

Training and computation in a deep neural net

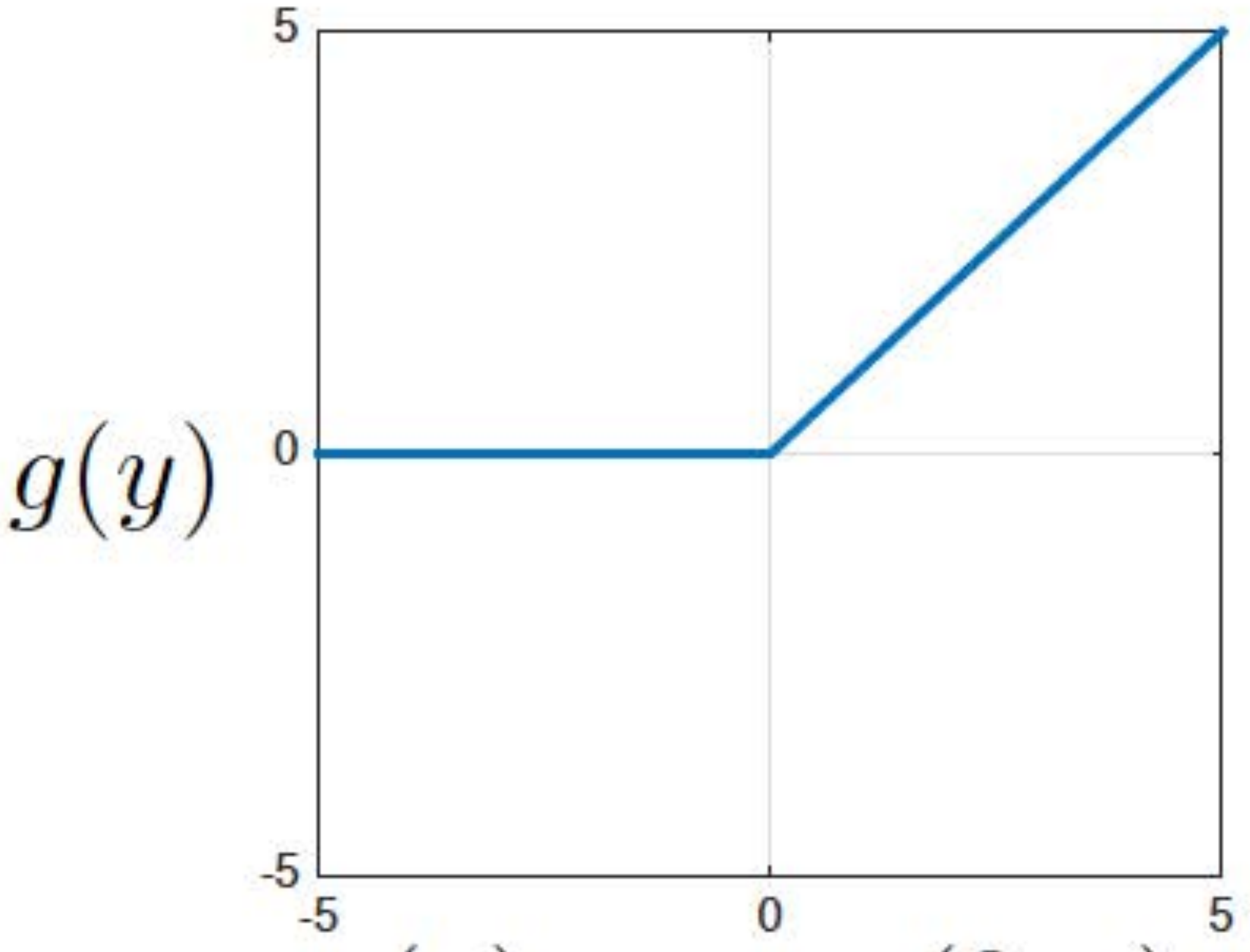
Computation in a neural net



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



Rectified linear unit (ReLU)



$$g(y) = \max(0, y)$$

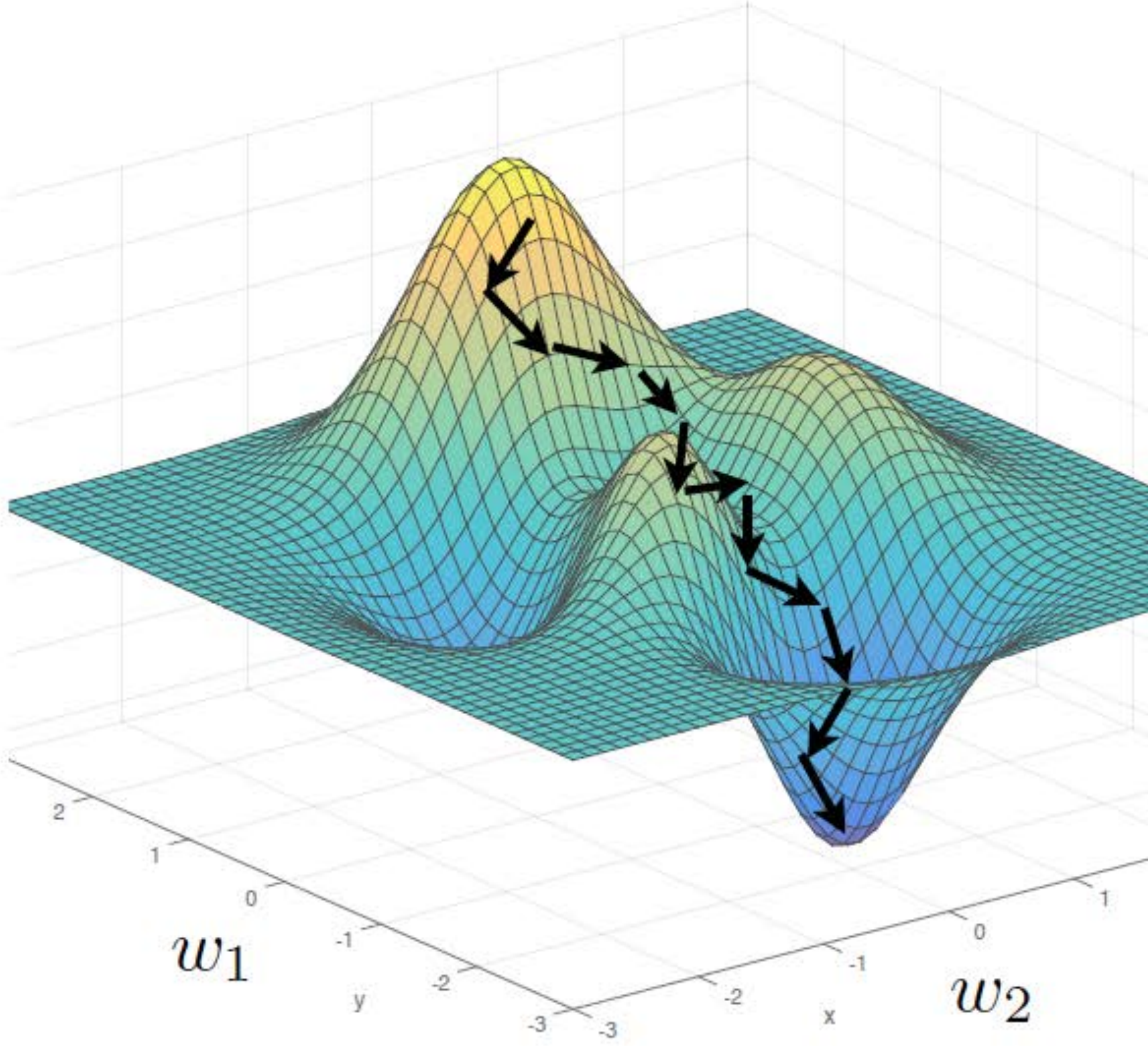
Gradient descent

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

One iteration of gradient descent:

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

learning rate



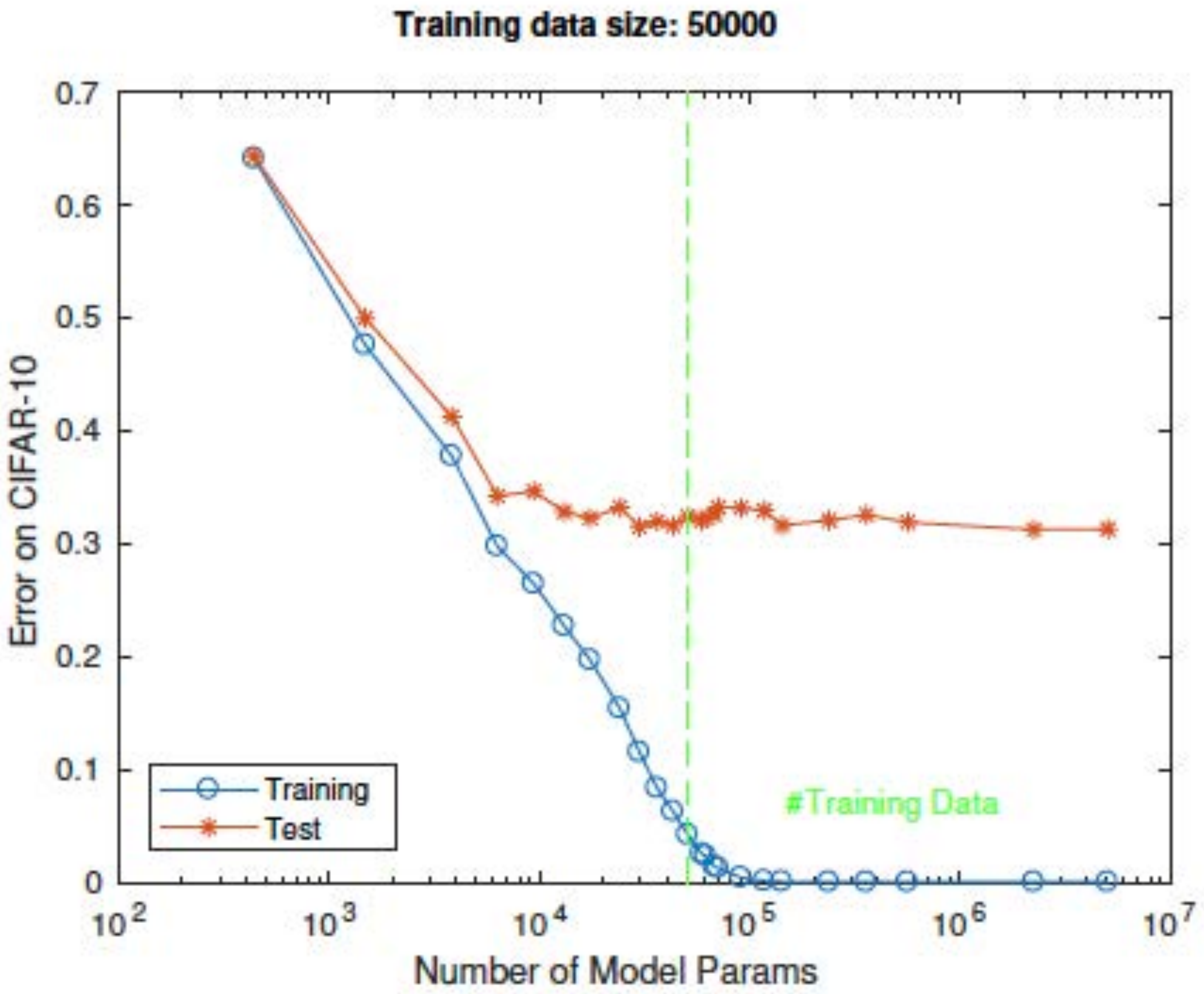
Course, part III, Deep Learning: theory questions

- why depth works
- why optimization works so nicely
- why deep networks do not overfit and do generalize

Deep nets : a theory is needed (after alchemy, chemistry)

Many reasons for this.
Today I will focus on bits of the puzzle
of good generalization
despite overfitting.

How can overparametrized solutions generalize?



How can deep networks generalize?
Where is the complexity control?

- The first observation is that classical learning theory has made clear that the number of parameters is not the key thing to be constrained. The norm of the parameters and related quantities such as VC dimension, Rademacher complexity, covering numbers are a better measure of complexity of the function that has to be controlled.

- You will see plenty of examples of this in the algorithms part of the course with regularization. You have seen the regularization term in one my slides.

- But deep nets have their overparametrization magic even without a regularization term (equivalent to weight decay) during training. Do we have something similar in classical math?

Classical algorithm:
Regularization in RKHS (eg. kernel machines)

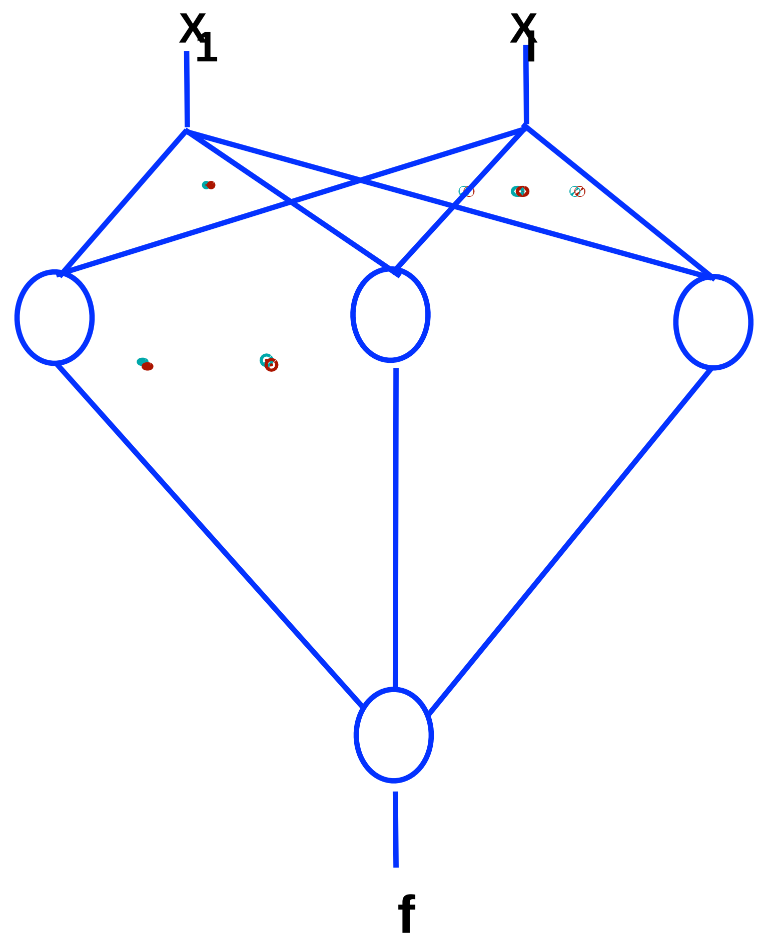
$$\min_{f \in H} \left[\frac{1}{n} \sum_{i=1}^{n} V(f(x_i) - y_i) + \lambda \, \|f\|_K^2 \right]$$

The regularization term controls the complexity of the function in terms of its RKHS norm

implies

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

Classical kernel machines — such as SVMs — correspond to <u>shallow</u> networks



Covering numbers and bits

A covering number is the number of spherical balls of a given size needed to completely cover (ε - net) a given space, with possible overlaps.

Example: The metric space is the Euclidean space, your parameter space K consists of d-dimensional vectors in the space with norm < R.

The covering numbers are $N_\varepsilon(K) = (\frac{2R\sqrt{d}}{\varepsilon})^d$

How can deep networks generalize?
Where is the complexity control?

- The first observation is that classical learning theory has made clear that the number of parameters is not the key thing to be constrained. The norm of the parameters and related quantities such as VC dimension and Rademacher complexity and covering numbers are a better measure to control.

- You will see plenty of examples of this in the algorithms part of the course with regularization. You have seen the term in one my slides.

- But deep nets have their overparametrization magic even without a regularization term (equivalent to weight decay) during training. Do we have something similar in classical math?

Pseudoinverse

One of the definitions of the Moore-Penrose pseudoinverse is

$$A^+ = \lim_{\delta \searrow 0} \left(A^* A + \delta I \right)^{-1} A^* = \lim_{\delta \searrow 0} A^* \left(A A^* + \delta I \right)^{-1}.$$

which can be seen (Lorenzo will explain in class 3) as the limit of a regularization λ going to zero.

Furthermore, when you do gradient descent on a linear network under the square loss, GD converges to the pseudoinverse if you start with close-to-zero weights (class 7).

CENTER FOR Brains Minds+ Machines

Unconstrained optimization of deep nets with exponential loss

Gradient descent on $L = \sum_{n}^{N} e^{-y_n f(W_K,\ldots,W_1;x_n)} = \sum_{n}^{N} e^{-y_n \rho\, \tilde{f}(V_K,\ldots,V_1;\, x_n)}$

gives the dynamical system

$$\dot{W}_k^{i,j} = -\frac{\partial L}{\partial W_k^{i,j}} = \sum^{N} e^{-y_n f(x_n)} y_n \frac{\partial f(x_n)}{\partial W_k^{i,j}}$$

which can be shown to be equivalent to

$$\dot{\rho}_k = \frac{\rho}{\rho_k} \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} \tilde{f}(x_n)$$

$$\dot{V}_k = \frac{\rho}{\rho_k^2} \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} \left(\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k V_k^T \frac{\partial \tilde{f}(x_n)}{\partial V_k}\right).$$

Unconstrained optimization of deep nets with exponential loss

The critical points of $\dot{V}_k$ are at finite ρ

$$\sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} = \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} V_k \tilde{f}(x_n)$$

Gradient descent on $L = \sum_{n}^{N} e^{-y_n f(W_K,\ldots,W_1;x_n)} = \sum_{n}^{N} e^{-y_n \rho\ f(V_K,\ldots,V_1;\, x_n)}$

gives a dynamical system with critical points for one effective support vector

$$V_k f(x_*) = \frac{\partial f(x_*)}{\partial V_k}$$

<u>*Constrained*</u> optimization of deep nets with exponential loss

Gradient descent on

$$L = \sum_{n}^{N} e^{-y_n \rho\, f(V_K,\ldots,V_1;\, x_n)} + \lambda_k \sum_k \left\|V_k\right\|^2$$

yields the dynamical system

$$\dot{\rho}_k = \frac{\rho}{\rho_k} \sum_{n}^{N} e^{-y_n \rho\, \tilde{f}(V_K,\ldots,V_1; x_n)} y_n \tilde{f}(x_n)$$

$$\dot{V}_k = \rho(t) \sum_{n}^{N} e^{-y_n \rho\, \tilde{f}(V_K,\ldots,V_1; x_n)} y_n \frac{\partial \tilde{f}(x_n)}{\partial V_k} - 2\lambda_k V_k \text{with}$$

$$\lambda_k = \frac{1}{2} \rho(t) \sum_{n}^{N} e^{-y_n \rho\, \tilde{f}(V_K,\ldots,V_1; x_n)} \tilde{f}(x_n)$$

Constrained optimization of deep nets with exponential loss

The critical points of $\dot{V}_k$ are at finite ρ

$$\sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} = \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} V_k \tilde{f}(x_n)$$

Gradient descent on $L = \sum_{n}^{N} e^{-y_n \rho \, f(V_K, \ldots, V_1; x_n)} + \lambda_k \sum_k \left\| V_k \right\|^2$

gives a dynamical system with critical points for one effective support vector

$$V_k f(x_*) = \frac{\partial f(x_*)}{\partial V_k}$$

Thus *constrained* and *unconstrained* optimization of deep nets with exponential loss by gradient descent correspond to dynamical systems with the same critical points at any finite time

Similarly to GD on a linear net under the square loss
GD here performs an implicit (vanishing) regularization. The underlying mechanism is different and more robust.

9.520/6.860

- classical regularization (regularized least squares, SVM, logistic regression, square and exponential loss), stochastic gradient methods, implicit regularization and minimum norm solutions. Regularization techniques, Kernel machines, batch and online supervised learning, sparsity.

- Classical concepts like generalization, uniform convergence and Rademacher complexities will be developed, together with topics such as surrogate loss functions for classification, bounds based on margin,, and pstabilityrivacy.

- Theoretical frameworks addressing three key puzzles in deep learning: approximation theory -- which functions can be represented more efficiently by deep networks than shallow networks-- optimization theory -- why can stochastic gradient descent easily find global minima -- and machine learning -- how generalization in deep networks used for classification can be explained in terms of complexity control implicit in gradient descent. It will also discusses connections with the architecture of the brain, which was the original inspiration of the layered local connectivity of modern networks and may provide ideas for future developments and revolutions in networks for learning.

Summary: the next breakthroughs

…are likely to come not from theory but from neuroscience…

Future >10y

NeoClassical

- Human Intelligence (HI) is memory based (exMachina)
- Depth is important for vision and other aspects of intelligence

➡We must find biologically plausible alternative to GD, perhaps layer-wise learning

➡We must find alternative to batch supervised learning, such as implicit labeling in time sequences

Scientific Revolution

- HI >>> memory
- Depth is misleading, not the norm, see mouse visual cortex

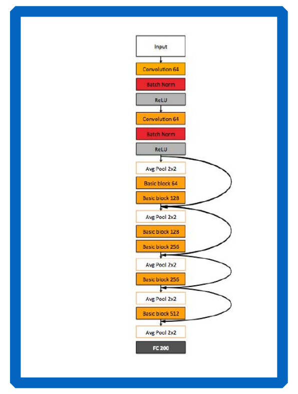
➡Thin recurrent networks=programs learned from time series

➡Cortex controls/manages routines

➡Evolution may have discovered programming early on…where is it in the brain?

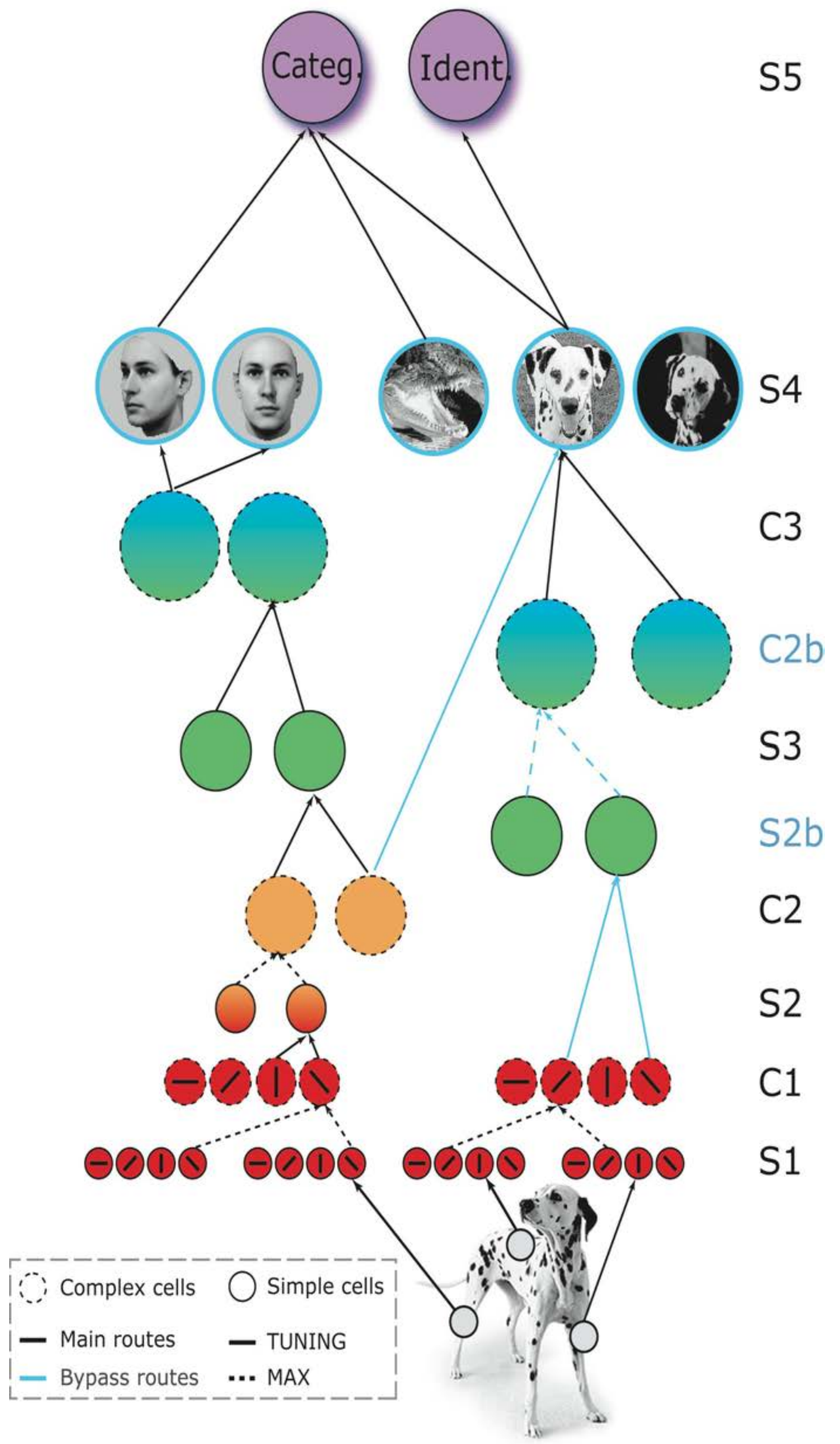
Musings on future progress (neoclassical)

- new architectures/class of applications from basic DCN block (example GAN + RL/DL + …)



- new semisupervised training framework, avoiding labels: implicit labeling…predicting next “frame”…

Musings on “revolutionary” Breakthroughs



Are deep nets really correct for biology? Is idea of depth misleading (look at the mouse visual system!)? Backprojection in multilayers is a biological pain! One layer recurrent machines are powerful!

General musings

The first phase of ML: supervised learning, big data $n \to \infty$

The next phase of ML: implicitly supervised learning, learning like children do, small data $n \to 1$

The evolution of computer science

- there were programmers
- there are now labelers, creating memory-based "intelligence"
- there will be bots who can learn like children do…