Unreasonable Effectiveness of Learning Artificial Neural Networks

Riccardo Zecchina

Politecnico di Torino & Human Genetics Foundation

Carlo Baldassi
Alessandro Ingrosso
Carlo Lucibello
Luca Saglietti

Politecnico di Torino
Human Genetics Foundation

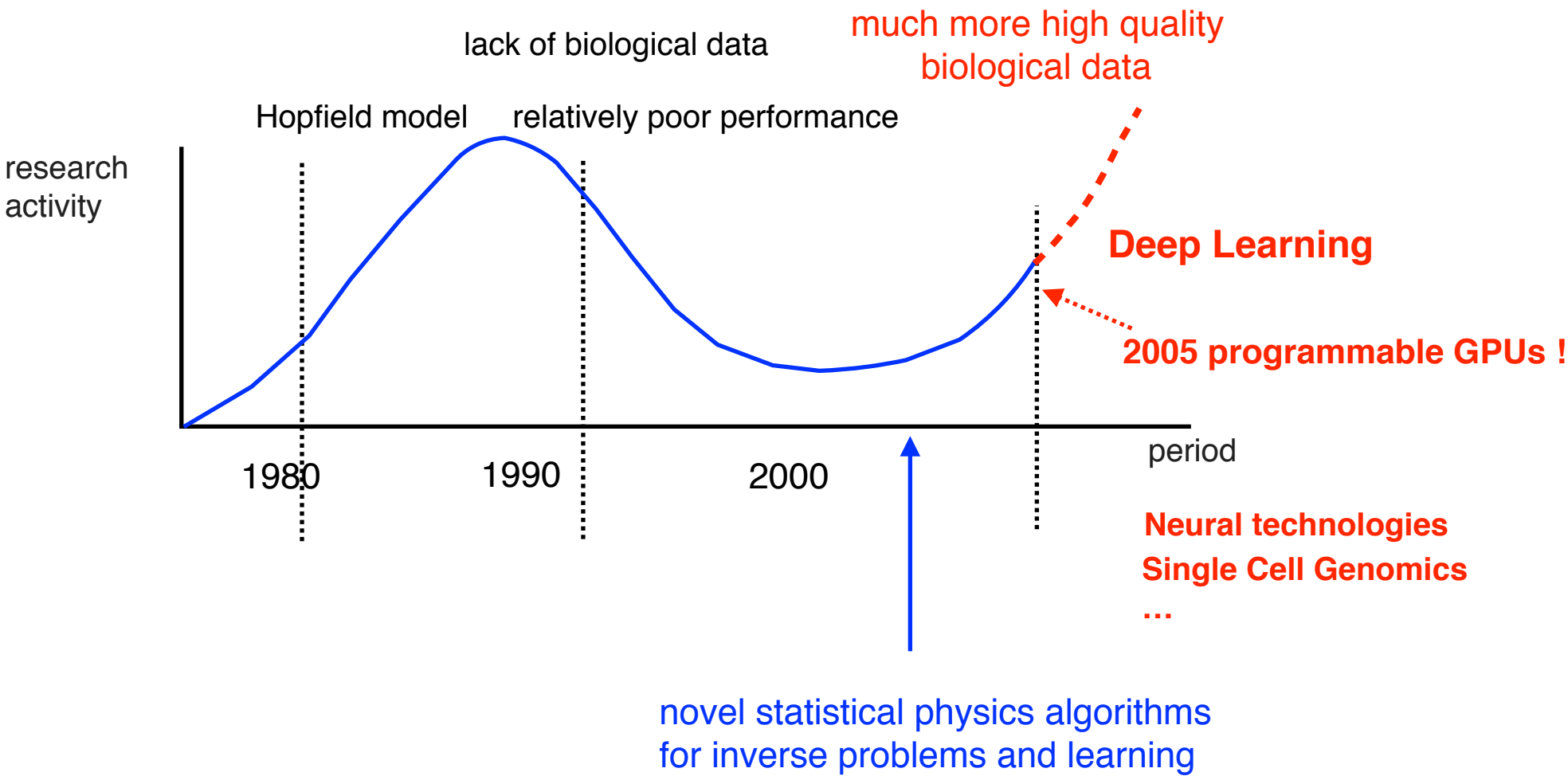
Christian Borgs
Jennifer Chayes,

Microsoft Research New England

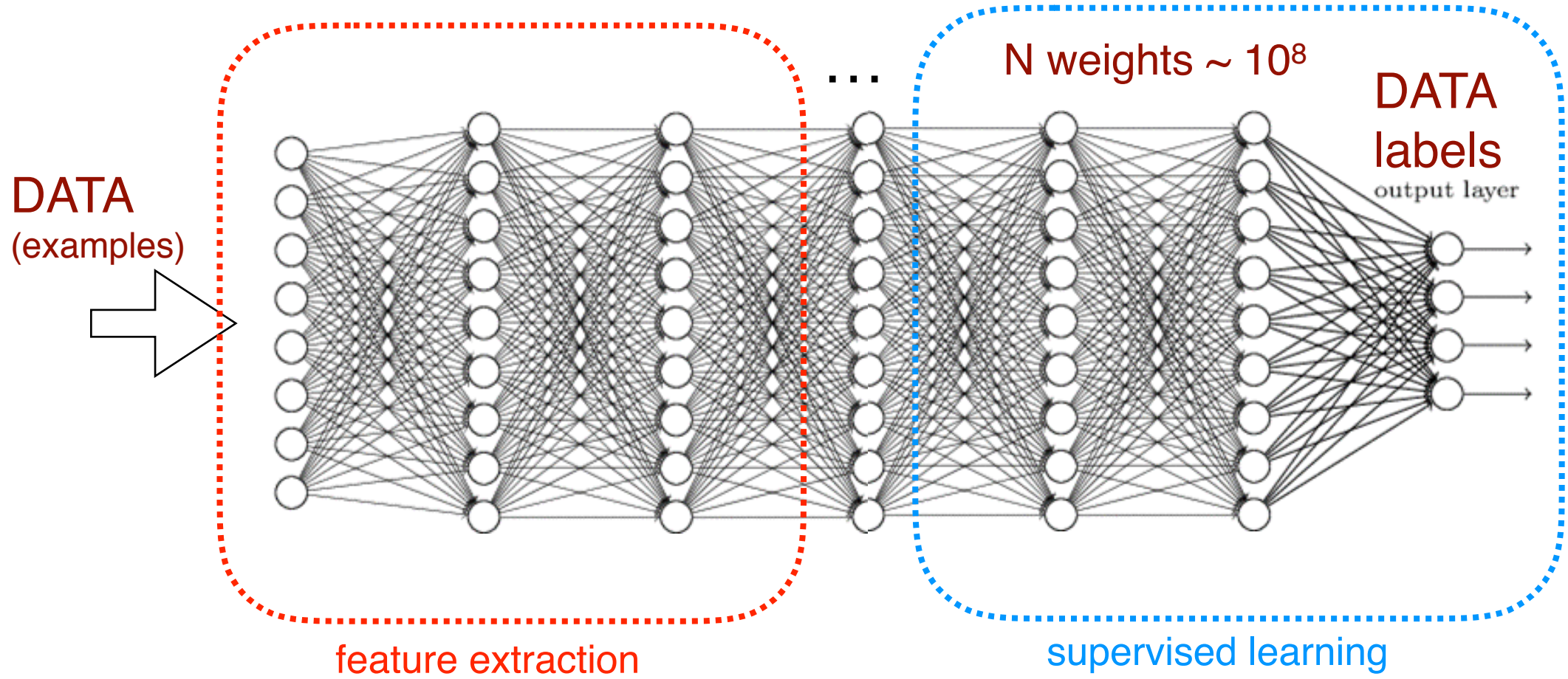
Plan of the talk

- Recent breakthroughs (and their limitations) in Machine Learning
- The case of Deep Learning
- Main questions:
  - » How is it possible to learn efficiently in highly non-linear artificial neural systems with 10^8-10^9 parameters (synapses)?
  - » How information is extracted?
- The next future

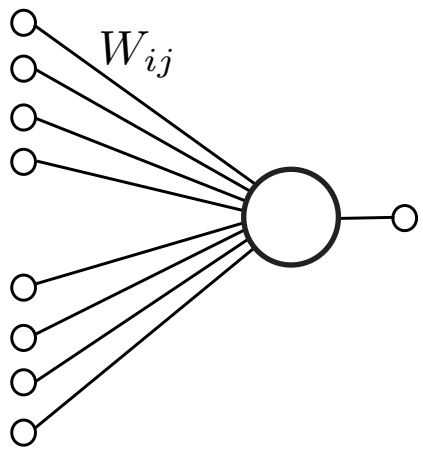
Computational neuroscience & artificial neural systems



Dissecting Deep Neural Networks



$\vec{\xi}^{\mu}$ ⇨ ⋮

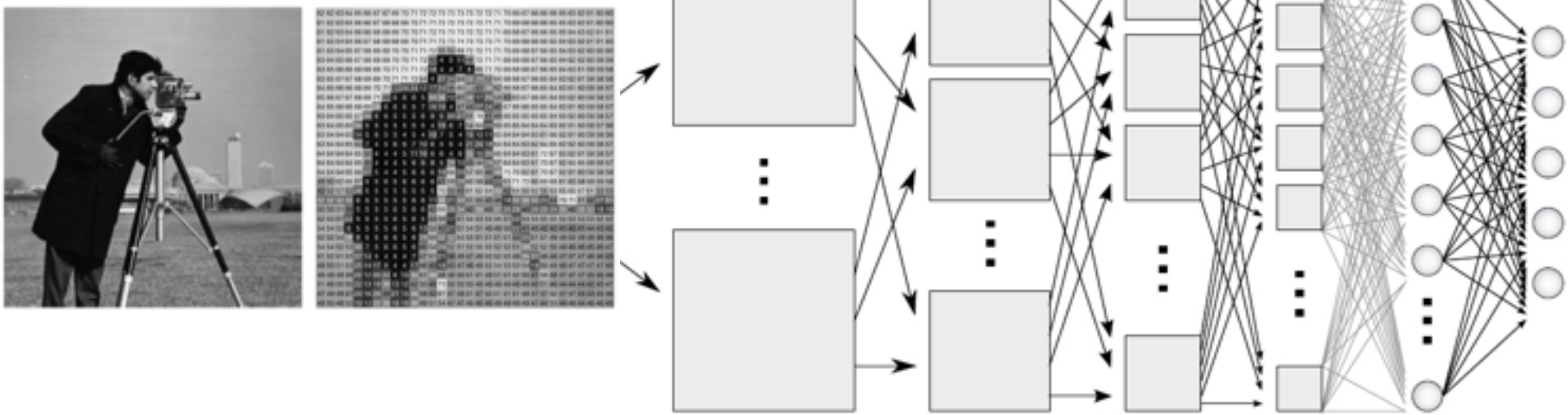


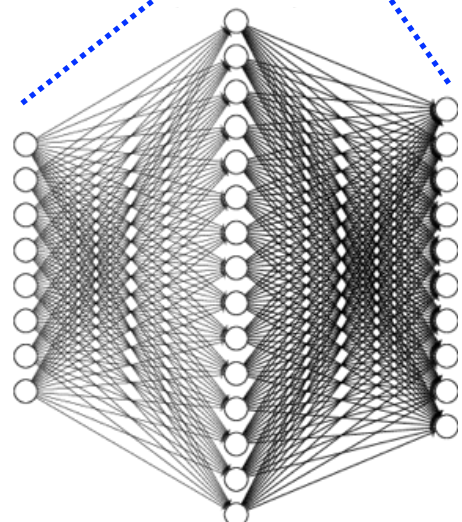
$$\sigma_j^{\ell+1} = f(\sum_i W_{ij}\sigma_j^{\ell})$$

(see demo)

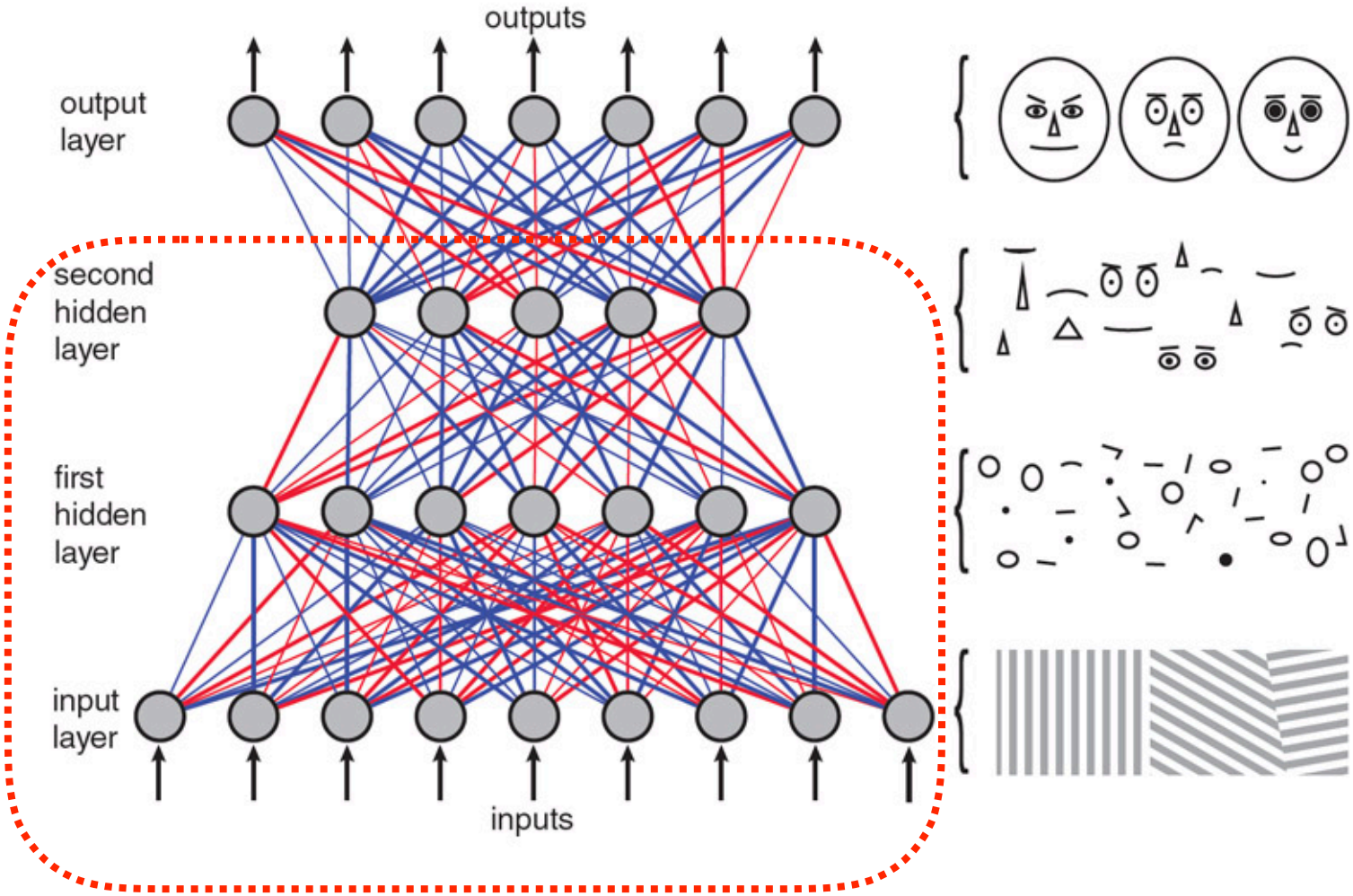
feature extraction

supervised learning:
(learning from examples)





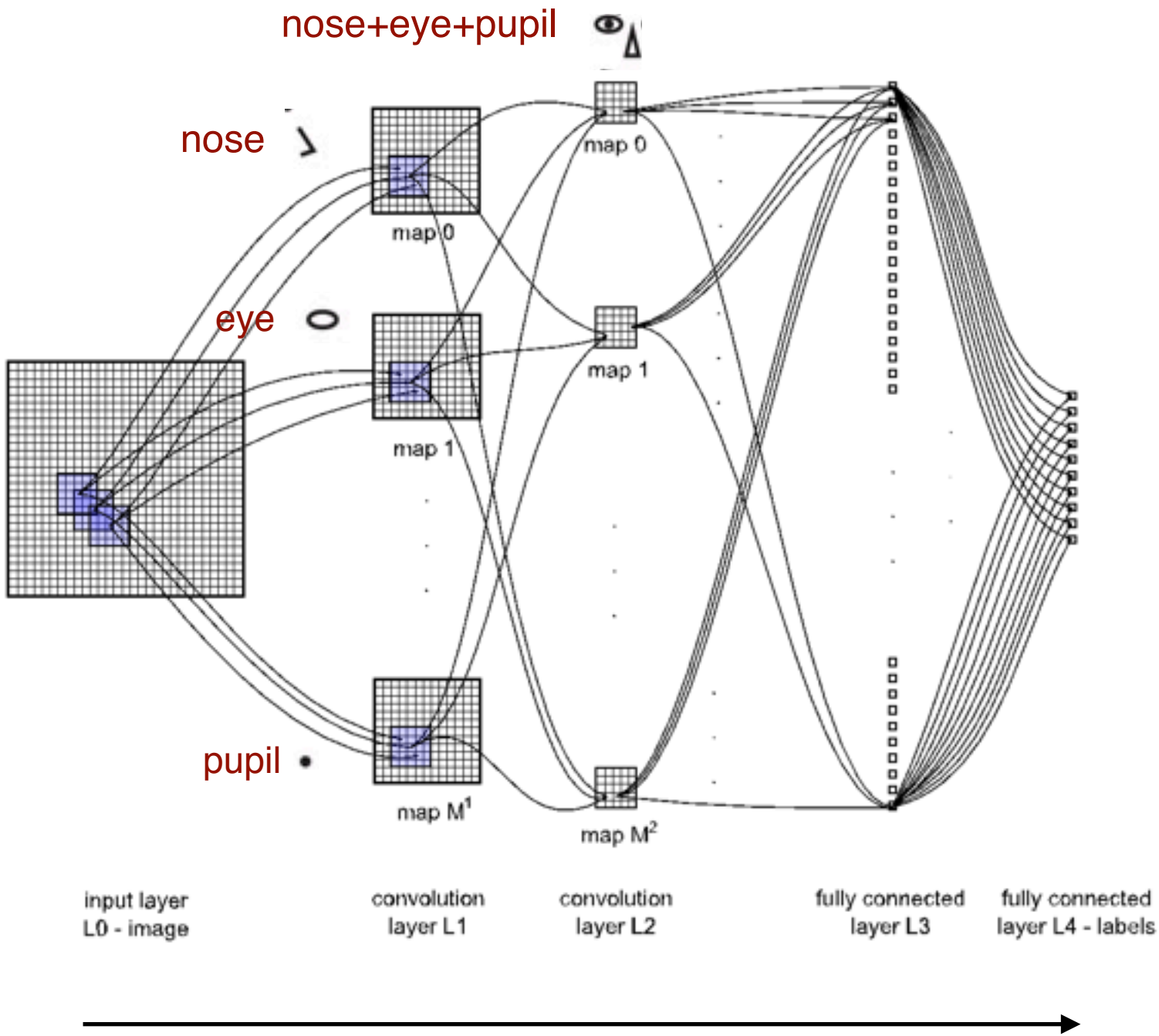
(vaguely inspired by the visual system)



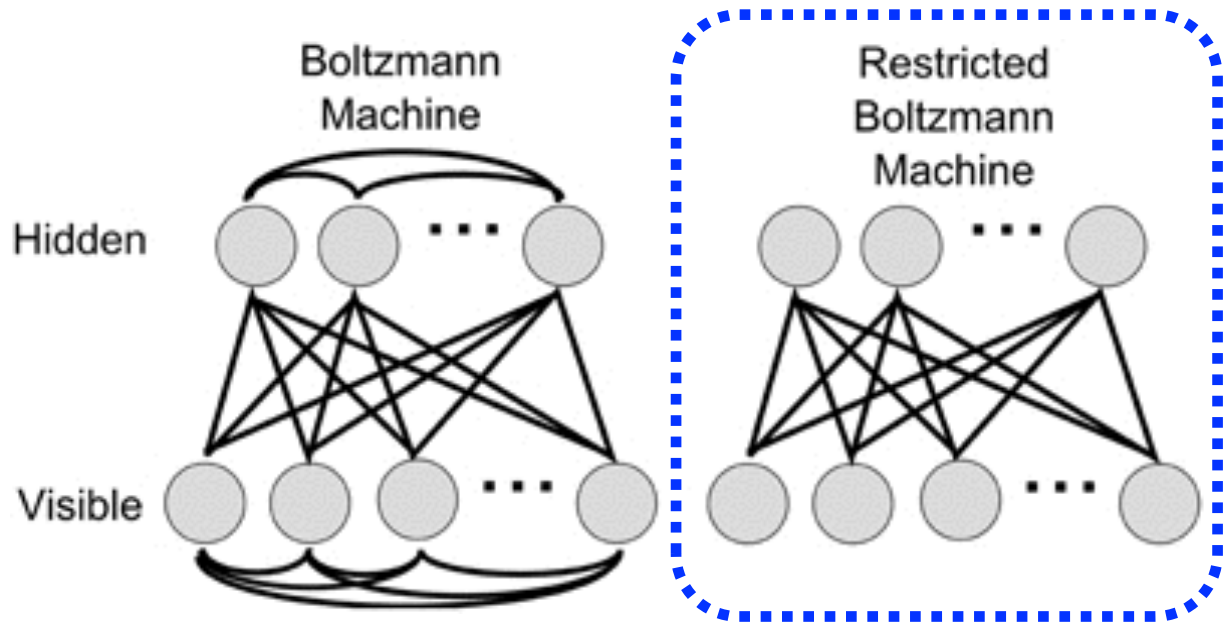
preprocessing is key

picture by Brian Hayes, 2009

Convolutional layers



Restricted Boltzmann Machines layers



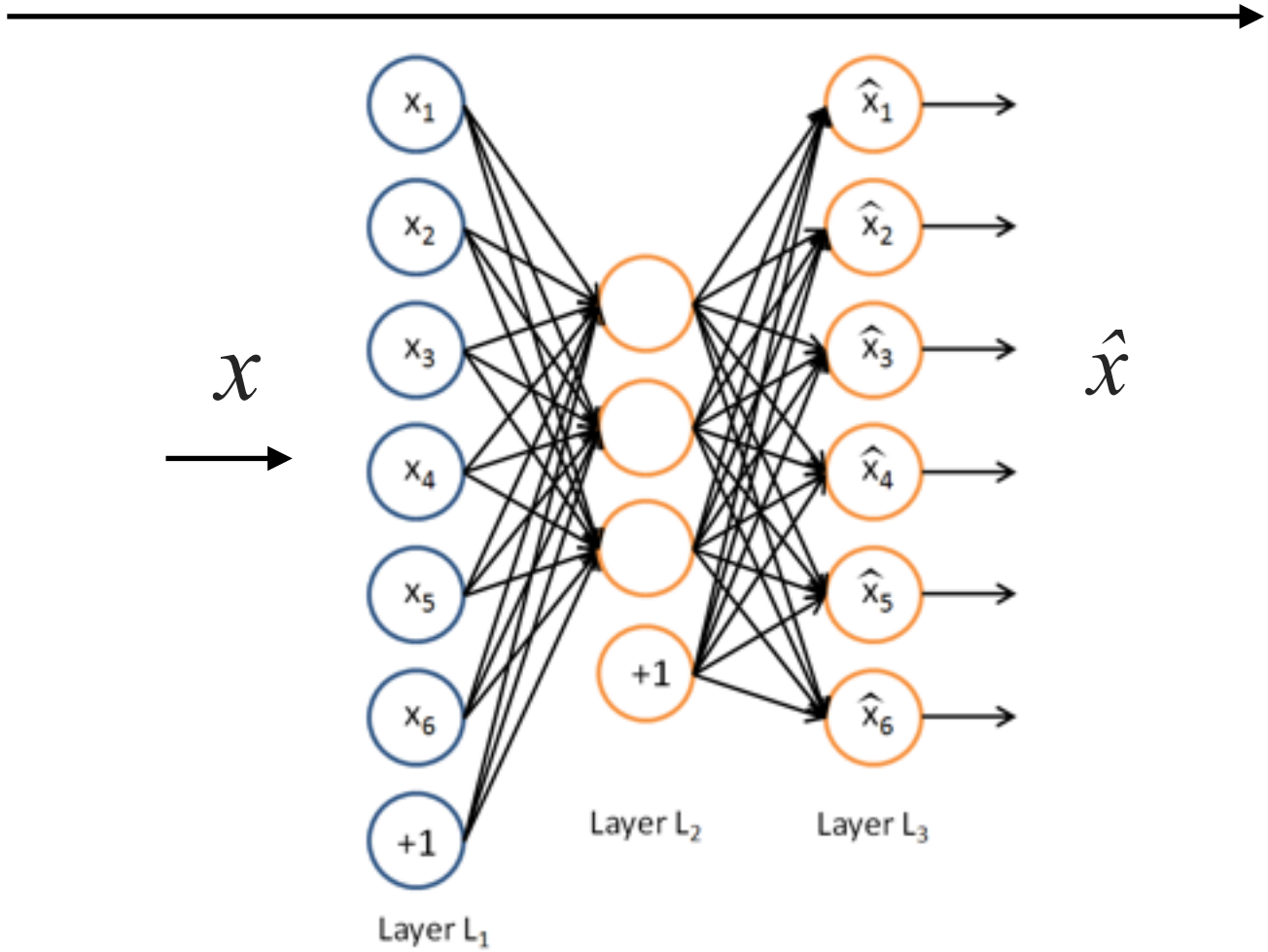
$$E(v,h) = -a^{\mathrm{T}}v - b^{\mathrm{T}}h - v^{\mathrm{T}}Wh$$

$$P(v) = \frac{1}{Z}\sum_{h} e^{-E(v,h)}$$

Training algorithm:

$$\arg\max_{W} \mathbb{E}\left[\sum_{v\in V} \log P(v)\right]$$

Auto-encoders layers: trained to *reconstruct* their own inputs



For each input x,

1. Forward pass: compute activations at all hidden layers and compute output $\hat{x}$
2. Measure the deviation of $\hat{x}$ from the input x
3. Backpropagate the error and perform weight updates.

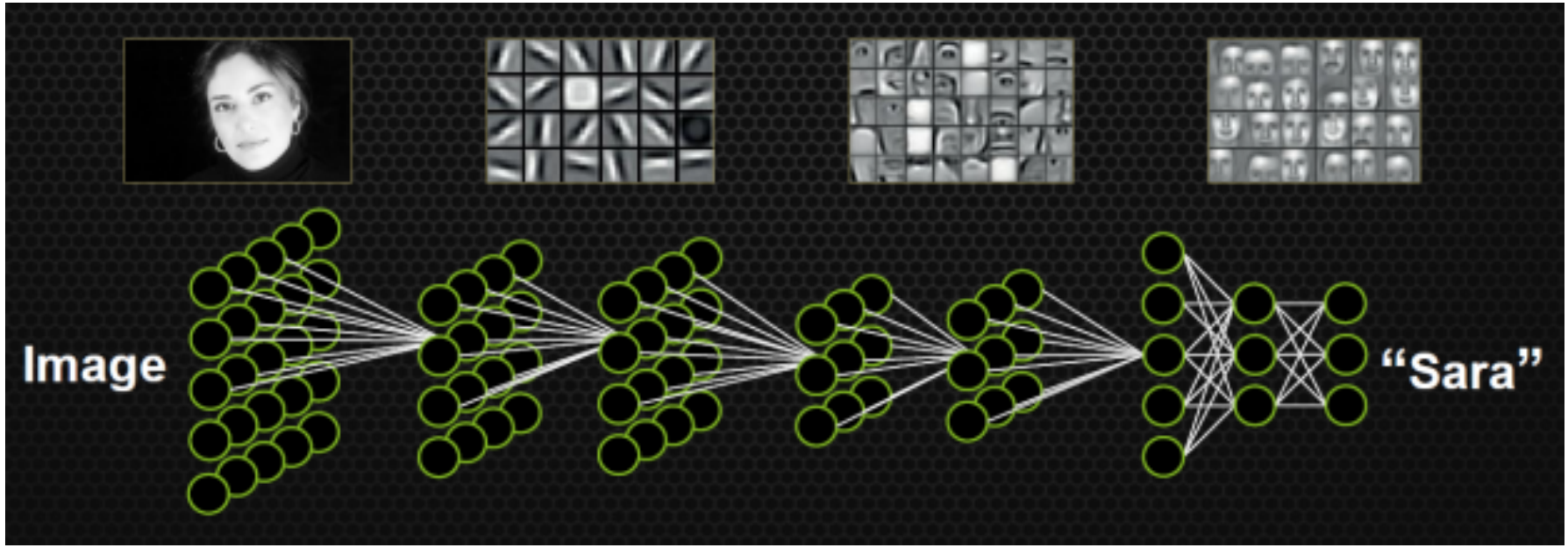
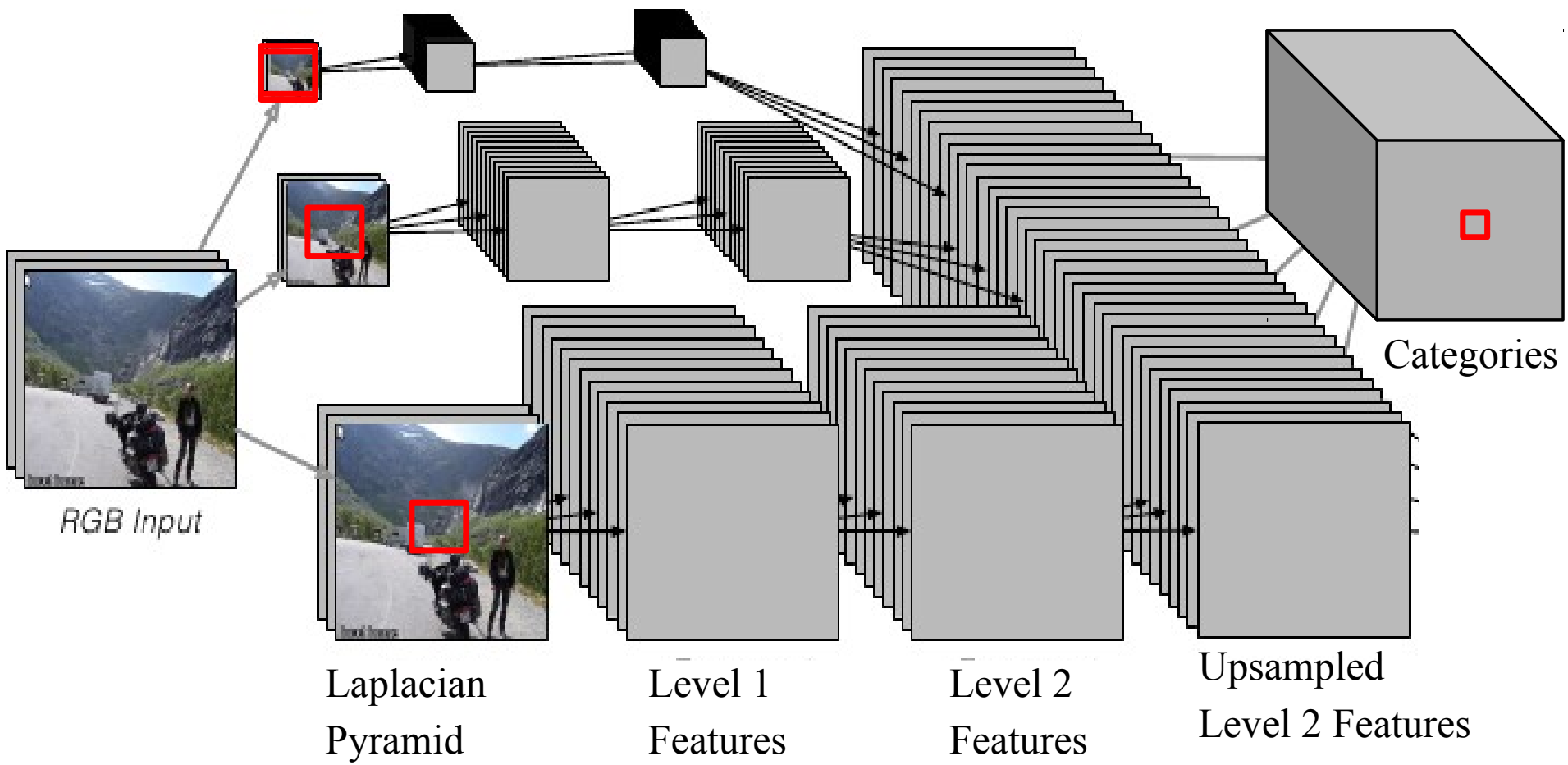
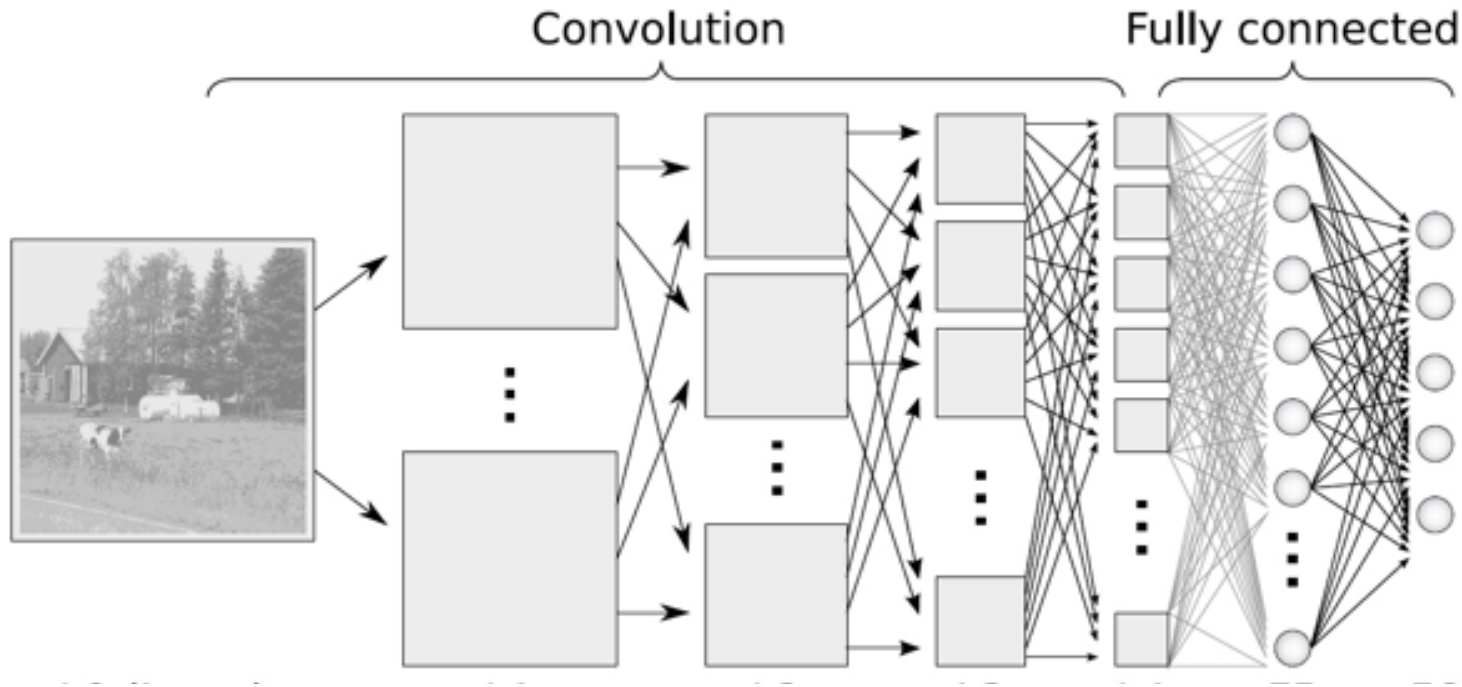


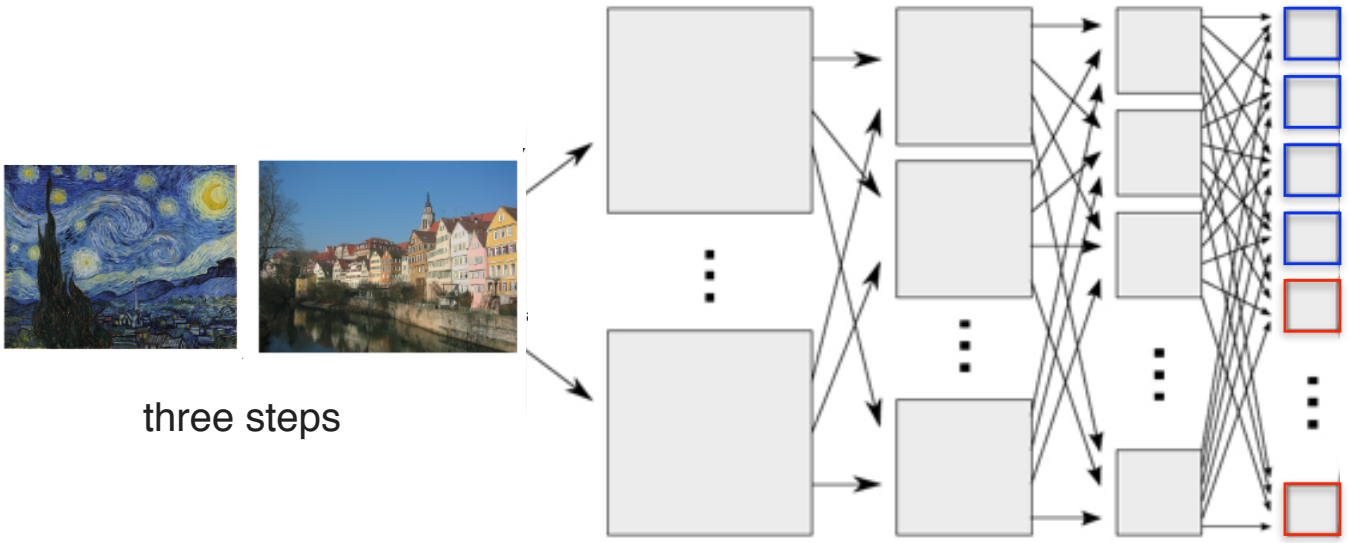
Image
"Sara"


RGB Input
Categories
Laplacian Pyramid
Level 1 Features
Level 2 Features
Upsampled Level 2 Features

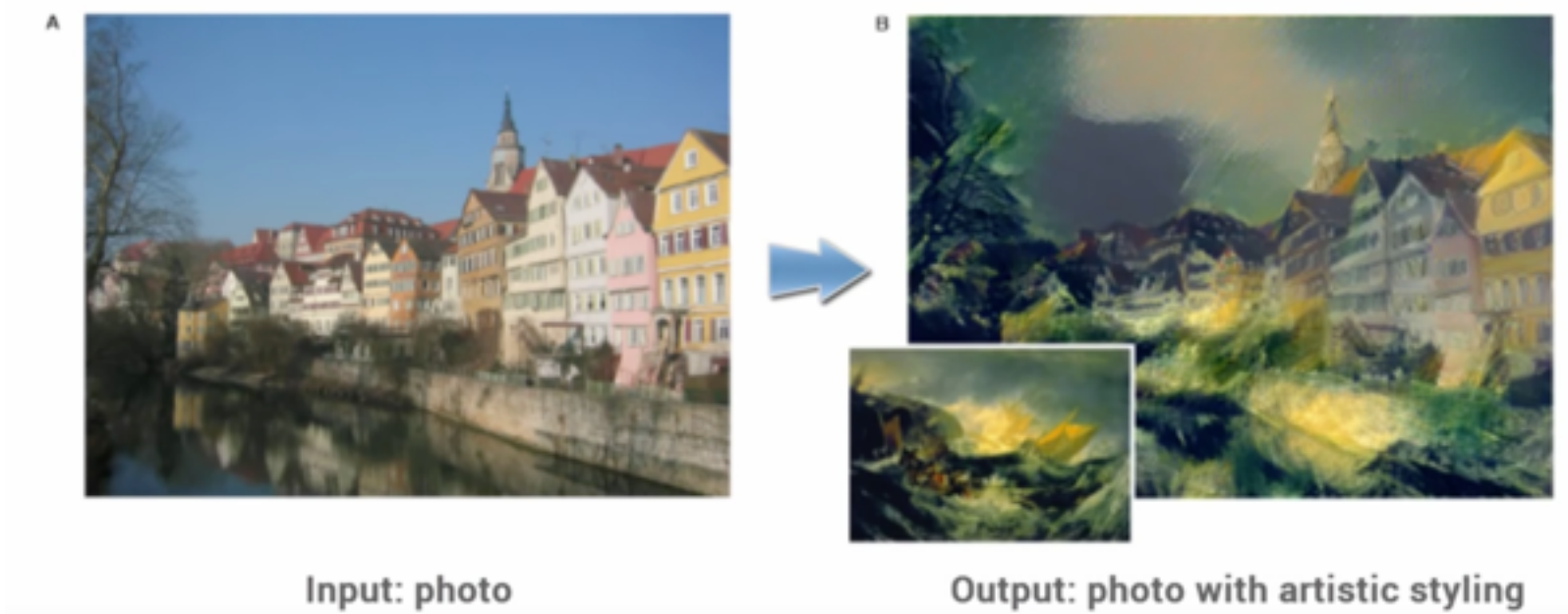


augmenting features

Fun with the first layers: pre-trained convolutional layers on huge image data sets



three steps



A Neural Algorithm of Artistic Style, L.A. Gatys, A.S. Ecker, M. Bethge , arXiv:1508.06576

A

F

A

D

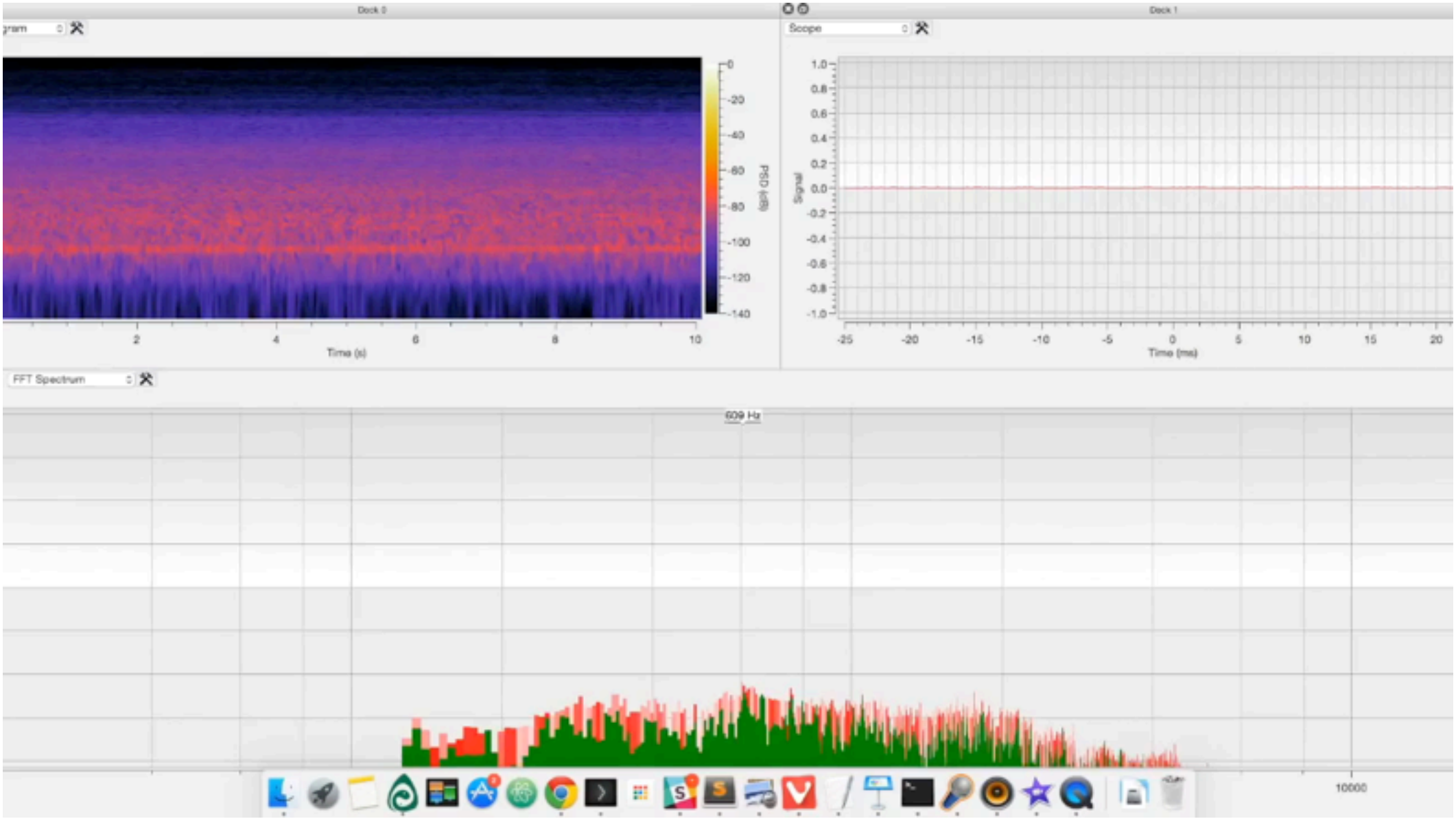
A

C

A

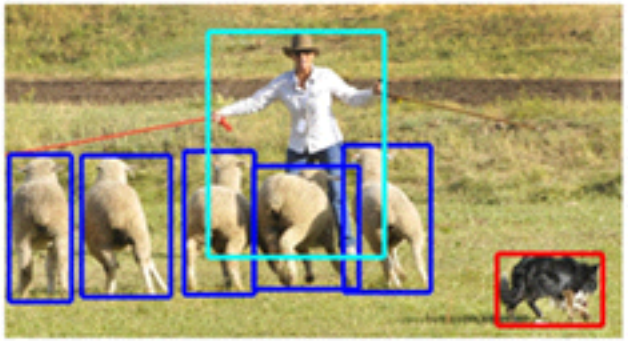
E

Speech recognition (Deep Speech, Baidu)



Deep translator

(a) classification



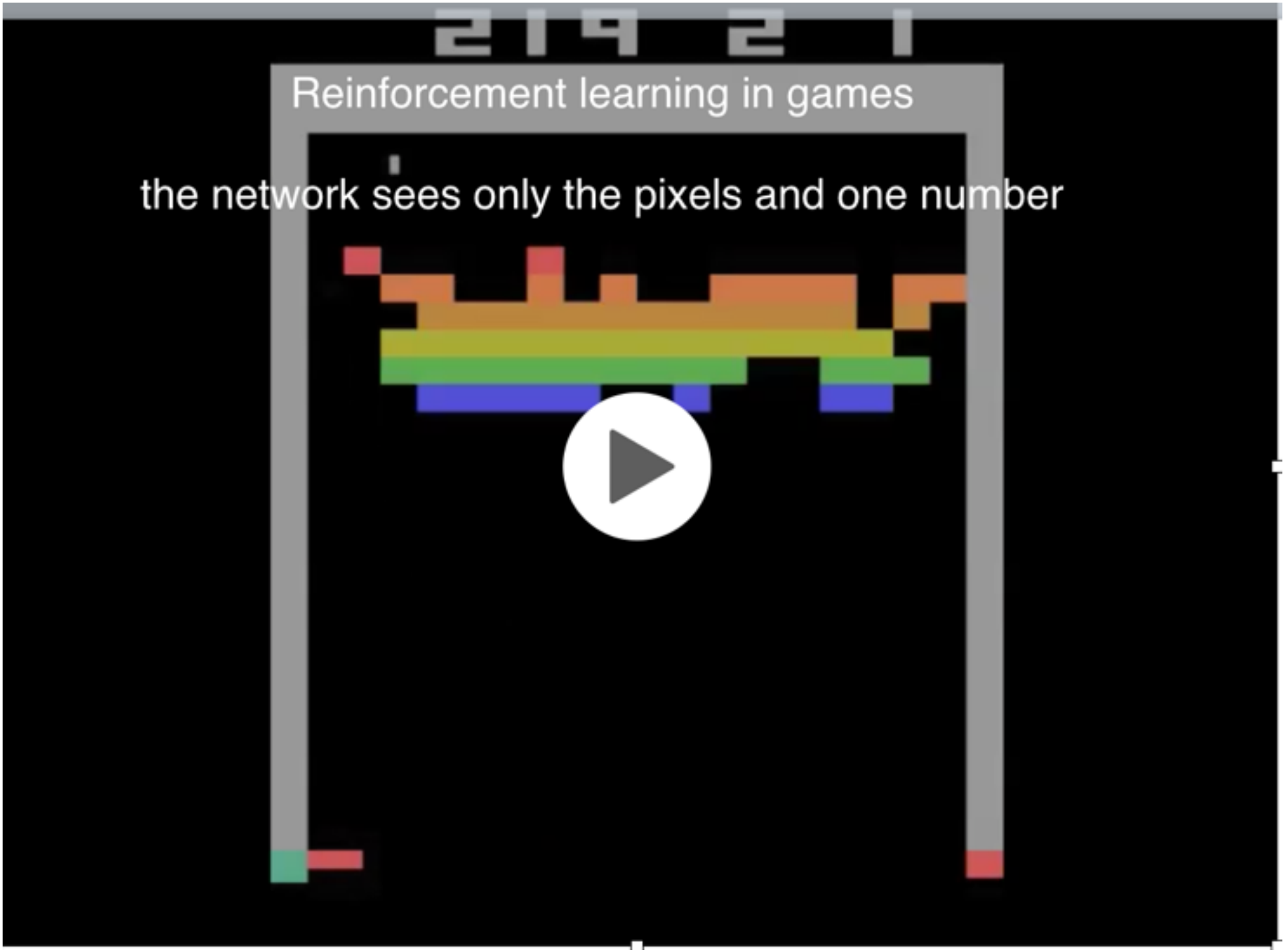
(b) detection

(c) segmentation

Current and short term applications

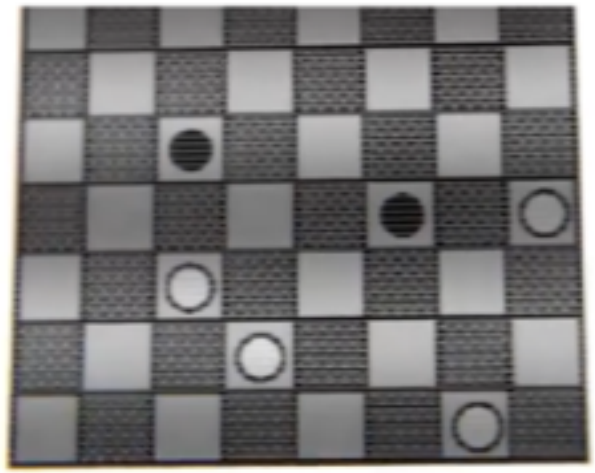
- Object recognition (e.g. explain pictures to visually impaired people)
- Speech recognition
- Data analysis in molecular biology
- Precision medicine
- Causal inference in economics (e.g. S. Athey, M. Taddy, …)
- Material science
- Web activities
- Image reconstruction in neuroscience
- …

(of course Deep Learning is just a component!)


Reinforcement learning in games
the network sees only the pixels and one number

human vs algorithms

1992

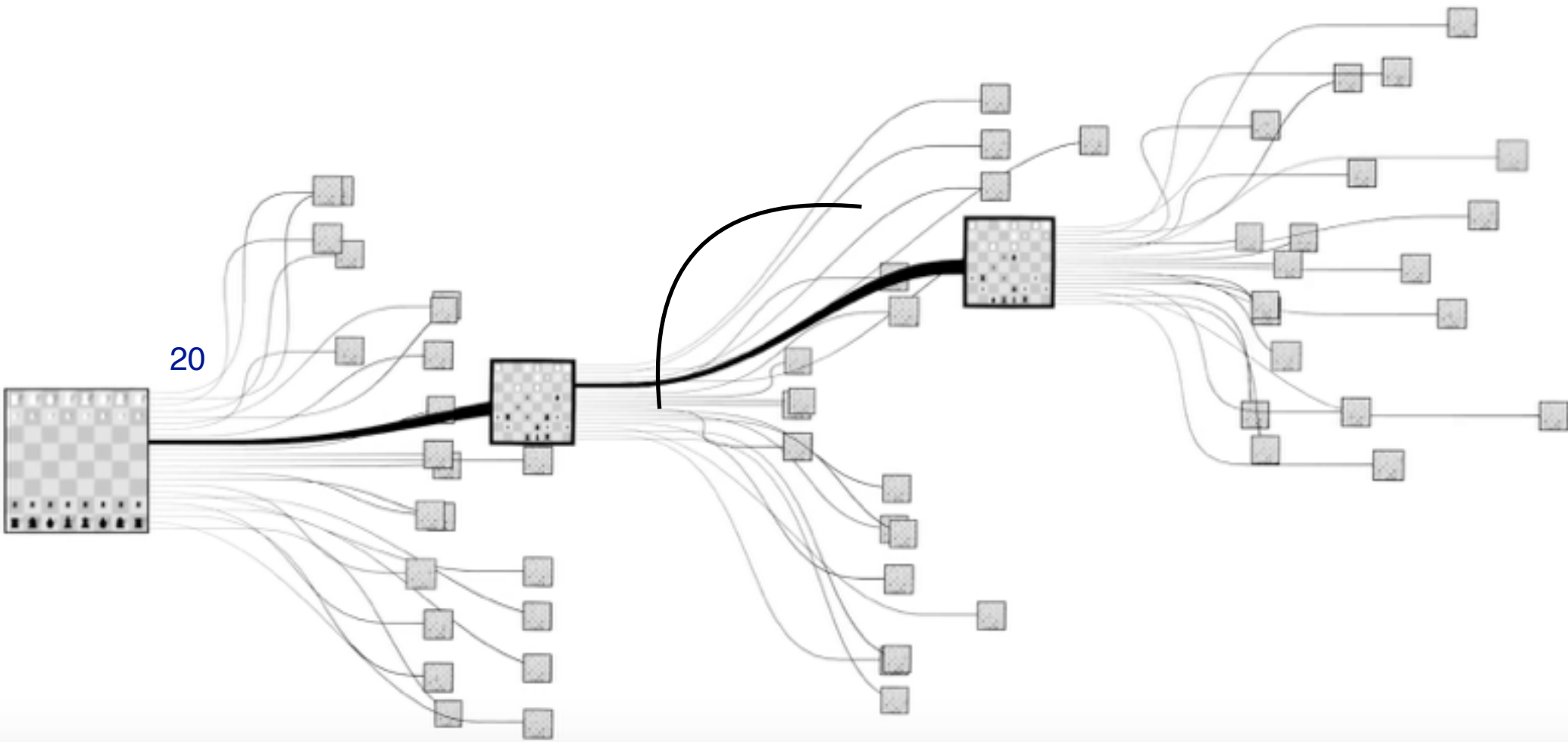


1994

1997

deep blue

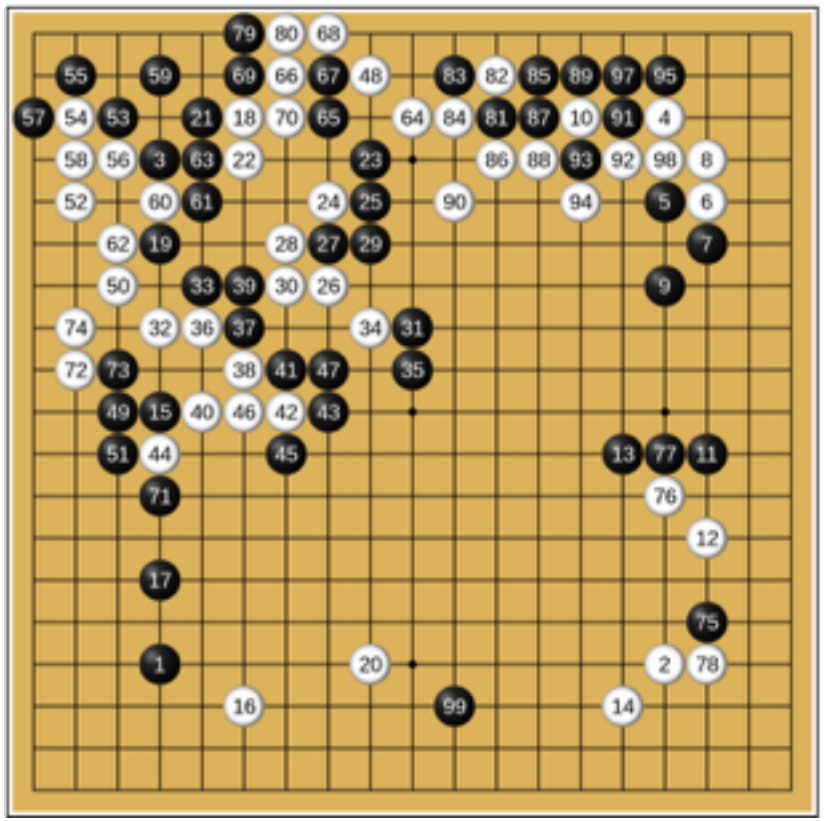
Search tree for Chess

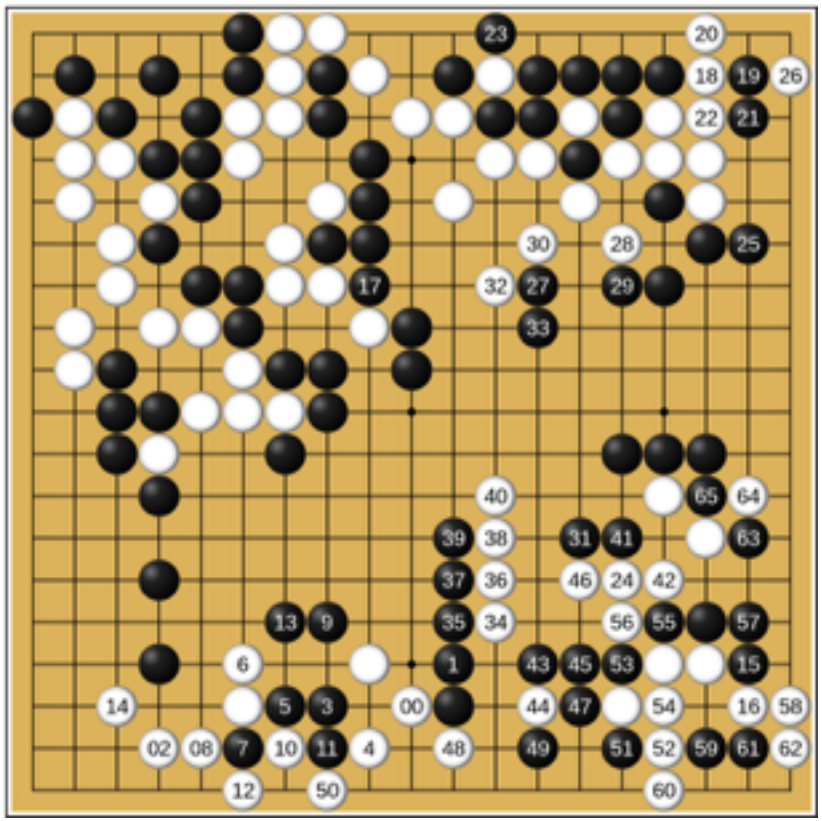


(today an app on your phone can beat any human player, Carlsen included, by optimally exploring the Chess tree)

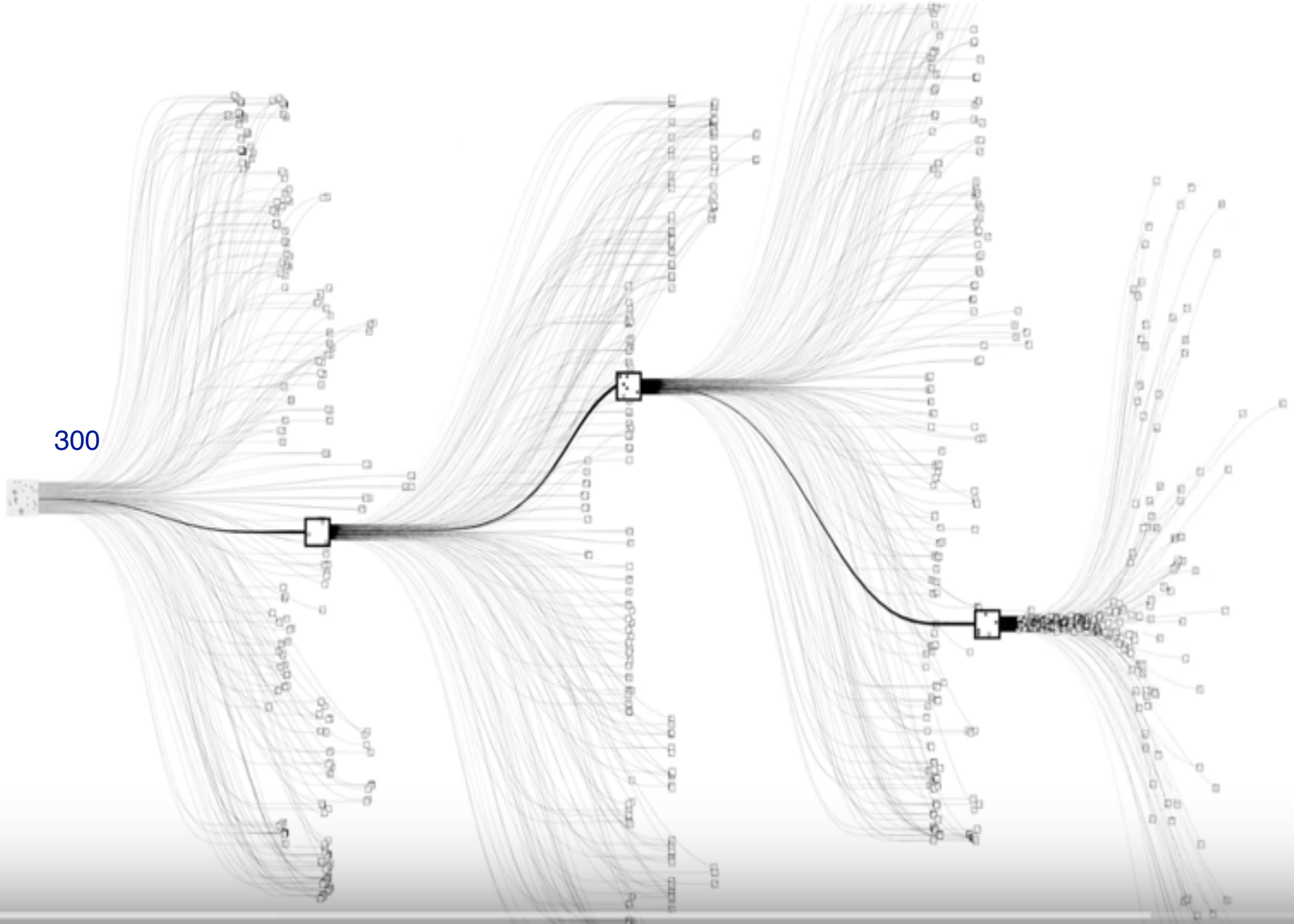
The game of GO: more configurations than atoms in the universe …

- perfect information (no luck)
- stones cannot move but can be captured if surrounded
- final objective: control 50% of the board

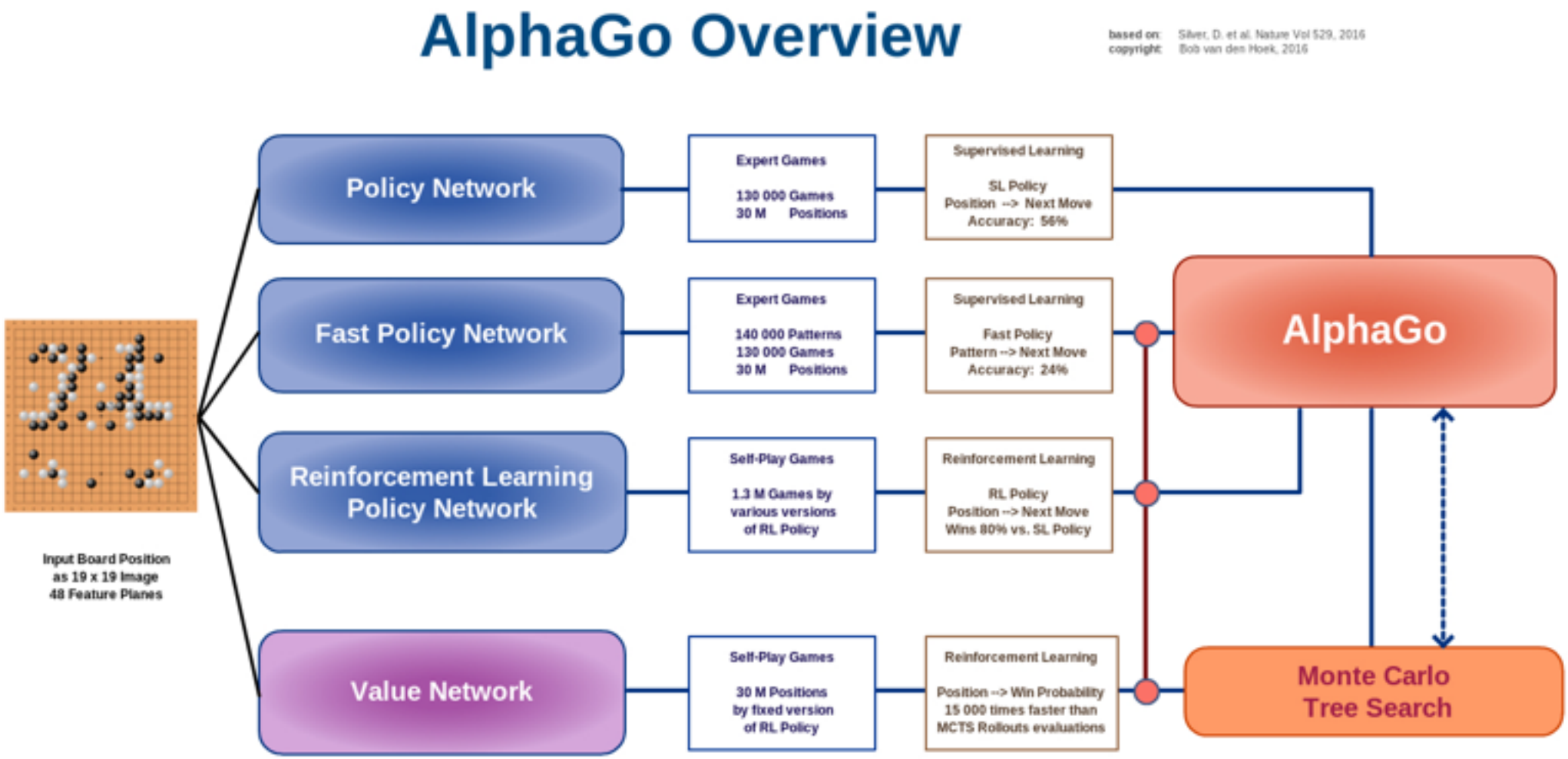




Search tree for Go



The search tree cannot be explored: intuitive game!



The search tree cannot be explored: intuitive game!

The search tree cannot be explored: intuitive game!

Next challenges in Machine Learning:

- *Differentiable computing*
- *Predictive learning: from unsupervised data to the ability to predict (in a given context)*

Bottlenecks:

- *Unsupervised learning*
- *Prediction under uncertainty (predict classes of outcomes)*

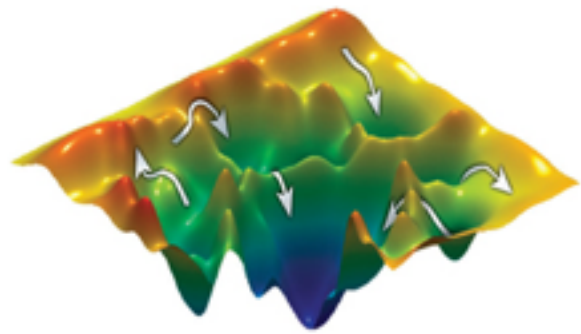
Still far away from AGI (20-40 years)

- *Building causal models*
- *Intuitive learning of physical or psychological phenomena*
- *transfer learning across different contexts*
- *….*

just to mention few

How does learning take place?

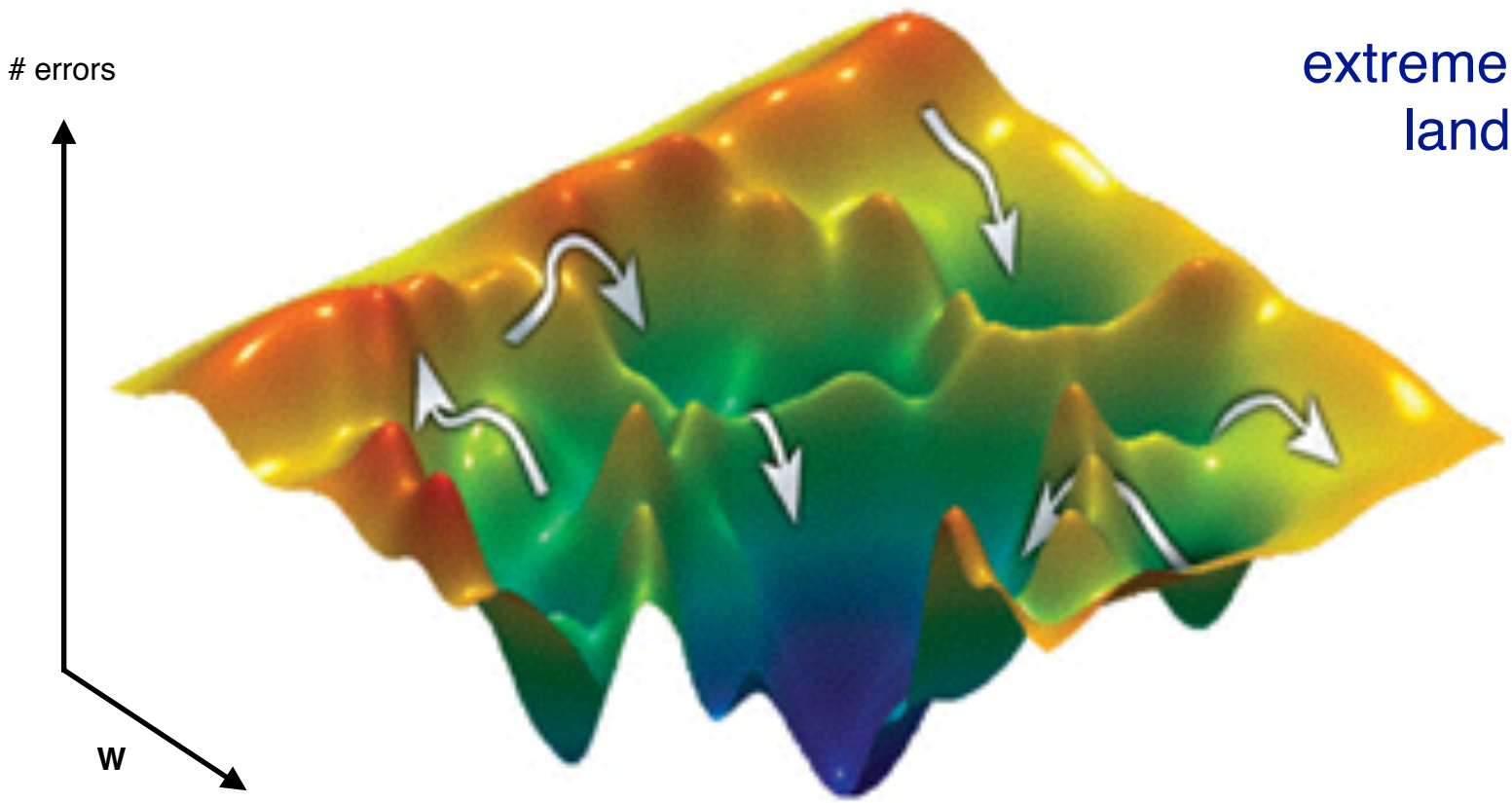
Learning algorithms: Variants of **gradient back-propagation** to minimise the number of errors (multitude of heuristic algorithms which have evolved in the last 30 years)



However there is a lack of theoretical understanding of when and why these algorithms work, even for artificial neural networks!

Learning ~ **energy minimisation problem** in high (10^{5}-10^{8}) dimension

$$H(\{W_{ij}^{\ell}\}) = \sharp \text{ errors} = \sum_{\mu} \Theta[-\sigma_{\mu} F(\{W_{ij}^{\ell}\})]$$



extremely complex landscape

However, successful algorithms do not “simply” minimize the energy.

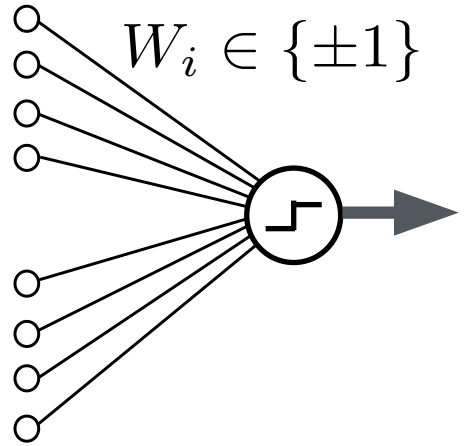
Kind of paradox! Why?

The simplest neural device: the Binary Neuron (Perceptron)

discrete “material” weights

$W_i \in \{\pm 1\} \quad , \quad i = 1, ..., N$

$\vec{\xi}^{\mu}$ ⇨ :



$$\sigma = \mathrm{sign}\left(\sum_{i=1}^{N} W_i \xi_i^{\mu}\right)$$

given a set of examples $\{\vec{\xi}^{\mu}, \sigma^{\mu}\}_{\mu=1,\ldots,P=\alpha N}$

find W such that $\sigma^{\mu} = \sigma(\mathbf{W}, \xi^{\mu}) \ \ \forall \mu$ ⇨ αN constraints on $\{W_i\}$

Minimize number of errors ~ minimum energy problem

$$H(\mathbf{W}) = \sum_{\mu} \Theta\left(-\sigma^{\mu} \mathrm{sgn}(\mathbf{W} \cdot \xi^{\mu})\right)$$ = # number of errors

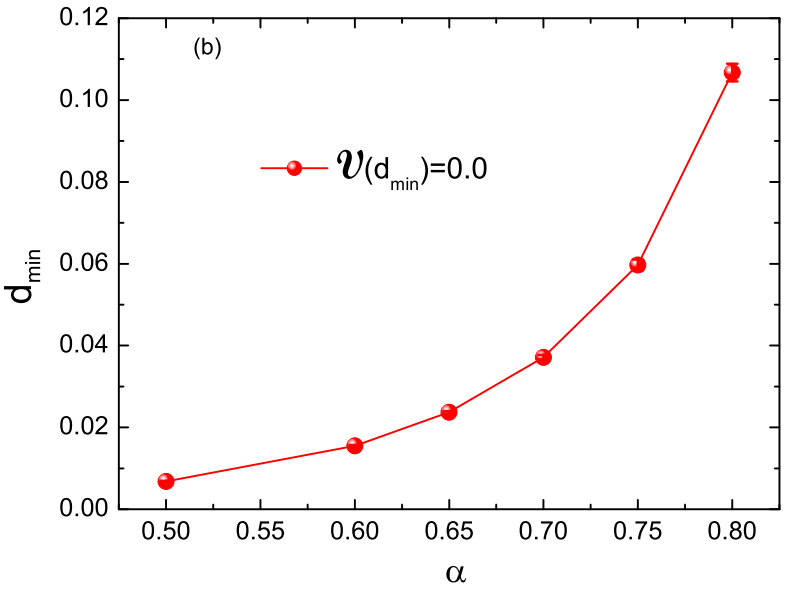
Geometry of the space of solutions

Typical distane between typical global minimal

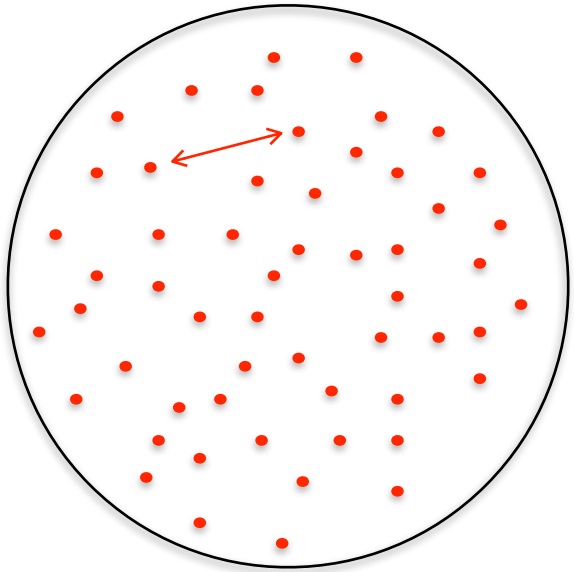
$$F(x) = \left\langle \frac{1}{Z(T')} \sum_{\mathbf{J}} \Theta\left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} J_i \xi_i^\mu\right) \ln \sum_{\mathbf{w}} \Theta\left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i \xi_i^\mu\right) e^{x \mathbf{J}\cdot\mathbf{w}} \right\rangle_{\boldsymbol{\xi}}$$

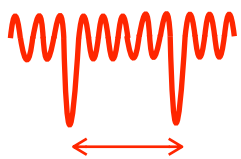
(Franz-Parisi potential)

$d_{\min}(\alpha)$



$d_{\min}(\alpha) \sim O(N)$





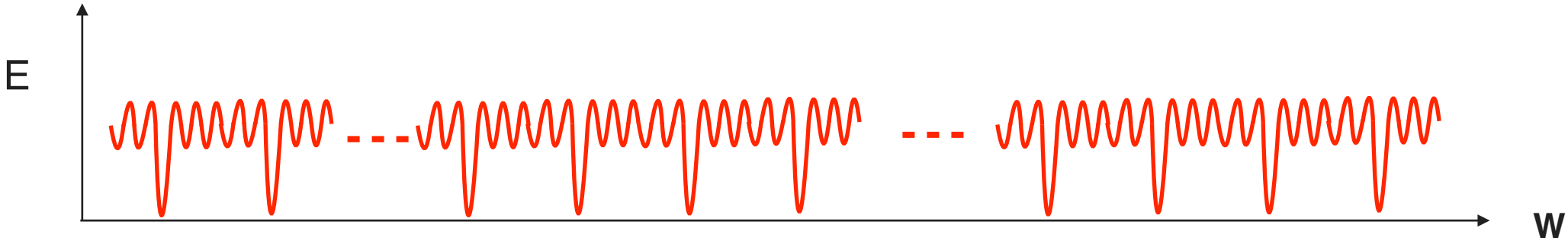
$d_{\min}(\alpha)$

H. Huang, Y. Kabashima (2014) (q_1=1 known since the 80's)

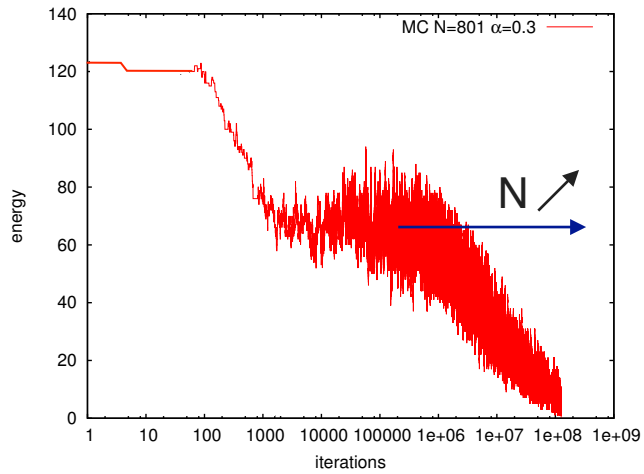
$$\alpha_c = \frac{P_{max}}{N} \simeq 0.83$$

W Krauth, M. Mezard, (1989)

Golf course energy landscape for any number of patterns !



Efficient learning impossible ?



But some algorithms have been found that work well!
How is this possible?

Local entropy measure & the Robust Ensemble

number of solutions within a distance d

$$\mathcal{N}\left(\tilde{W},d\right)=\sum_{\{W\}}\mathbb{X}_{\xi}\left(W\right)\delta\left(W\cdot\tilde{W},N\left(1-2d\right)\right)$$

where $\mathbb{X}_{\xi}\left(W\right)=\prod_{\mu=1}^{\alpha N}\Theta\left(\sigma^{\mu}\tau\left(W,\xi^{\mu}\right)\right)$

"energy" = local entropy

$$\mathcal{E}_d(\tilde{W})\doteq-\log\mathcal{N}(\tilde{W},d)$$

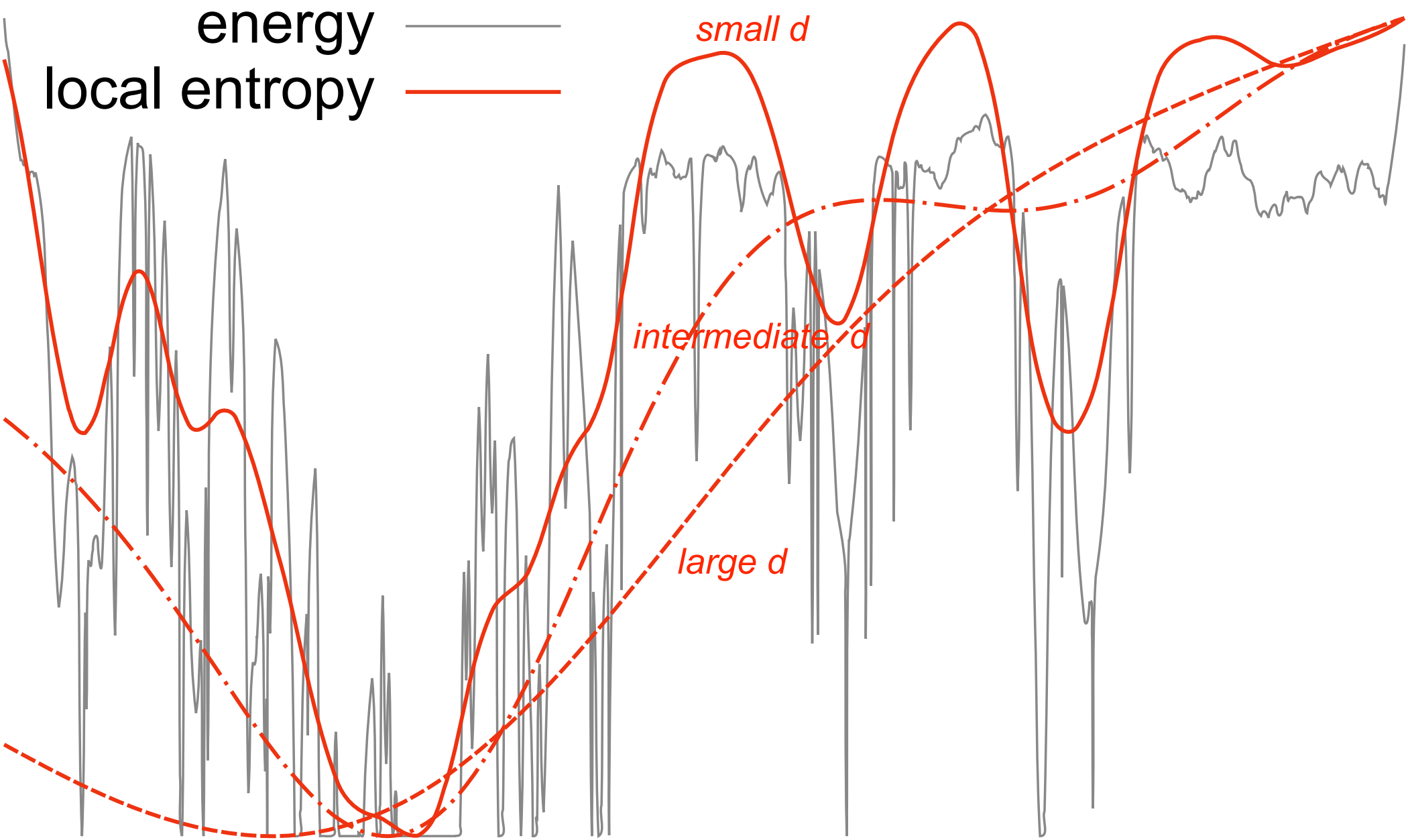
Robust Ensemble

$$\mathcal{P}(\tilde{W})\propto e^{-y\mathcal{E}_d(\tilde{W})}$$

maximally dense cluster $y\to\infty$

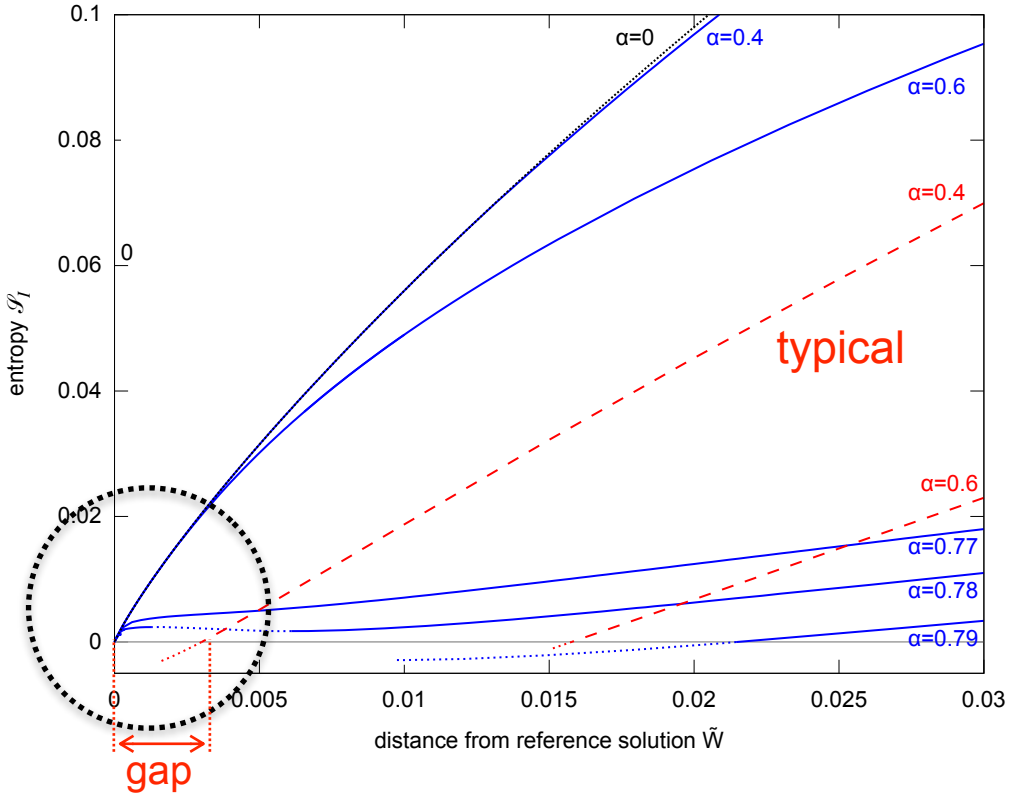
normalisation

$$Z(d)=\sum_{\{\tilde{W}\}}X_{\xi}(\tilde{W})e^{-y\mathcal{E}_d(\tilde{W})}$$

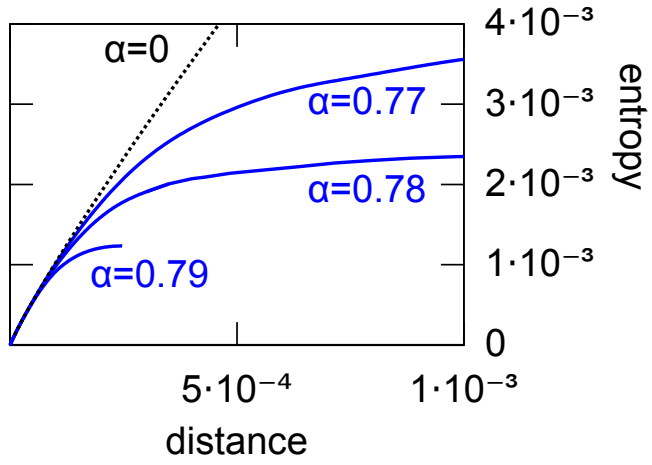


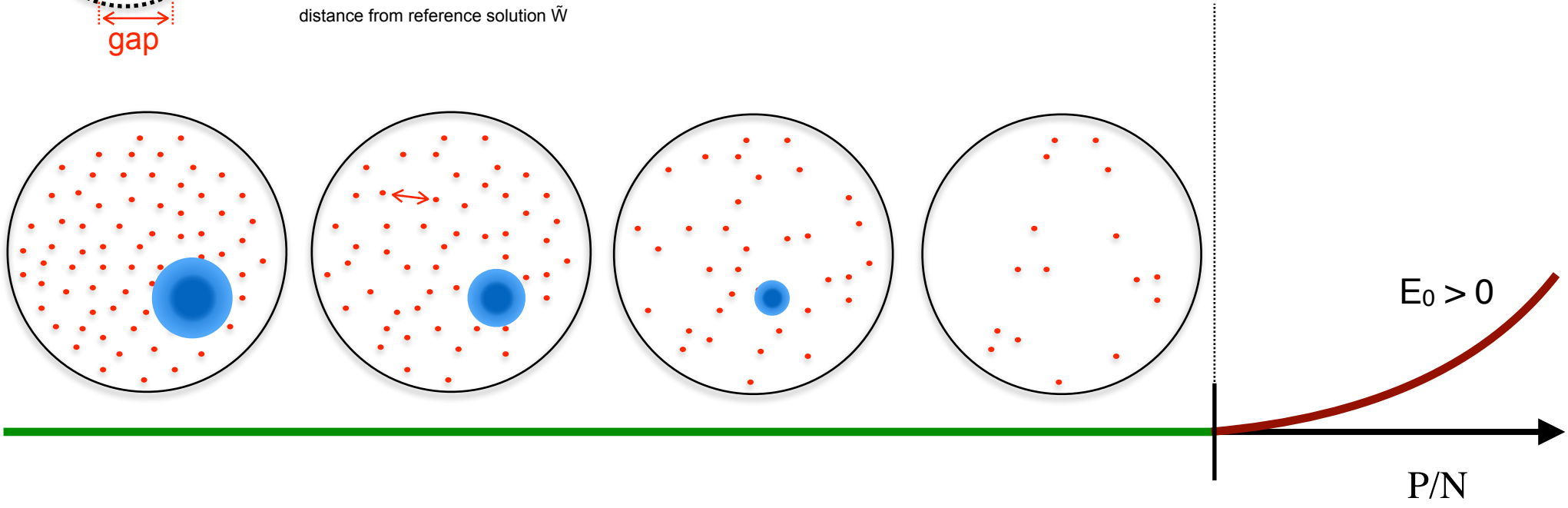
C. Baldassi, C. Borgs, J. Chayes, A.Ingrosso, C. Lucibello, L. Saglietti, RZ, PNAS 2016

Local entropy analysis of the binary perceptron



ultra-dense cluster





Principled algorithm:
Local Entropy driven MCMC

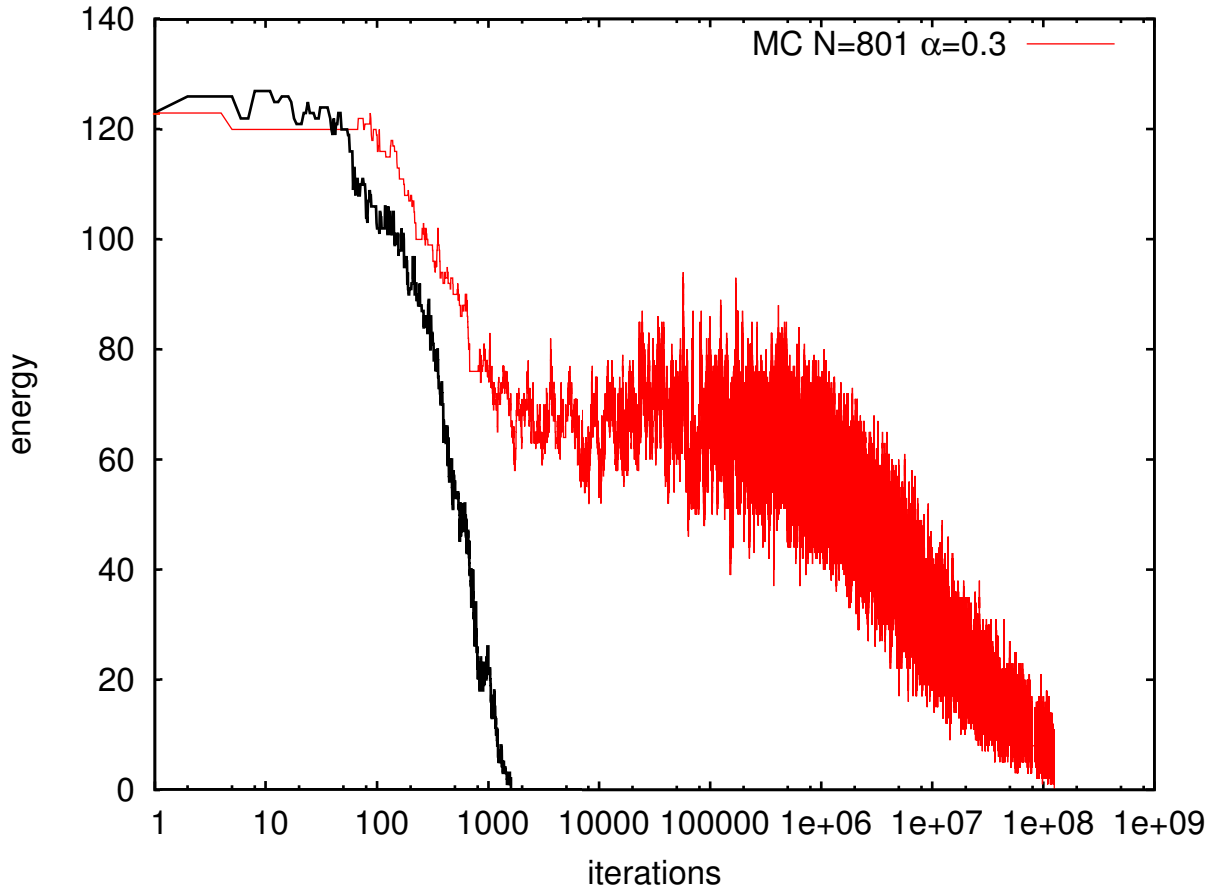
Objective Function:
search for configurations which maximize the local entropy

$$\mathscr{E}\left(\tilde{W}\right) = -\log \mathcal{N}\left(\tilde{W}, d\right)$$

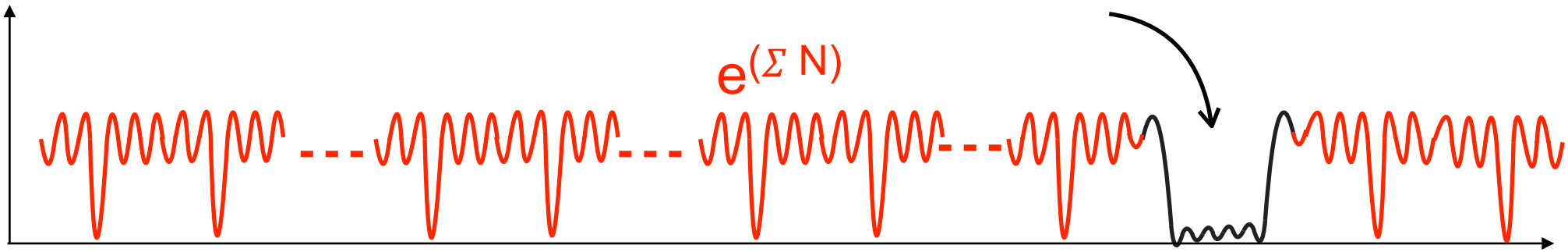
1. SA moves

2. BP to compute the local entropy

Local entropy driven stochastic search



$$\mathcal{P}_{\rm Gibbs}(C) \propto e^{-\frac{1}{T}E(C)}$$



Conclusions

Accessible sub-dominant states ~ Learning ~ out-of-equilibrium physics

- Entire new classes of learning algorithms with good Bayesian prediction capabilities (PNAS, 2016)
- Few bits of precision per synapse are indeed sufficient (PRL, PRE 2015)
- Role of accessible states in other systems/problems.
- Accessible dense states are targeted by state of the art DNN algorithms (2016)
- Applications in unsupervised learning

The next big acceleration?

On chip learning, large scale systems and data