Learning Deep Architectures for AI

Yoshua Bengio

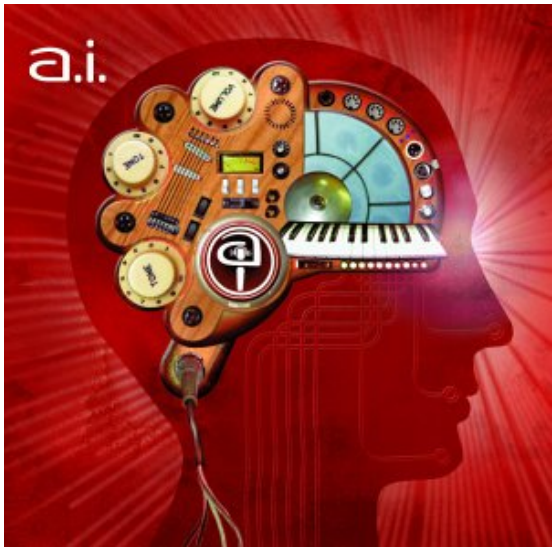
May 2008

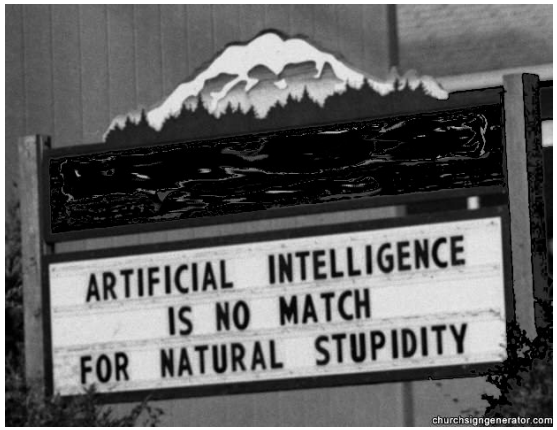
Thanks to : James Bergstra, Olivier Breuleux, Aaron Courville, Olivier Delalleau, Dumitru Erhan, Pascal Lamblin, Hugo Larochelle, Jerome Louradour, Nicolas Le Roux, Pierre-Antoine Manzagol, Dan Popovici, François Rivest, Joseph Turian, Pascal Vincent

Check review paper (with same title) on my web page

Artificial Intelligence

- Half century of research and goal seems still so far
- Why ?
- Too much in a hurry for *results* rather than *understanding* ?

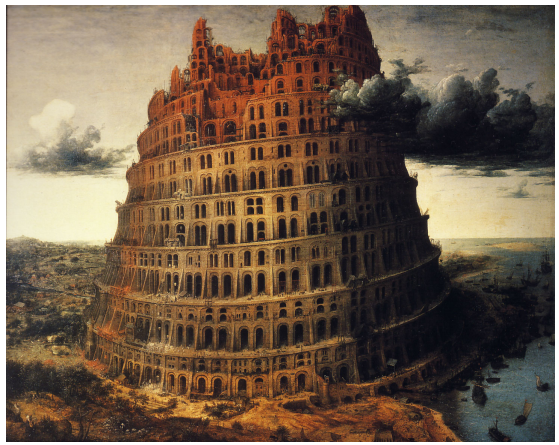




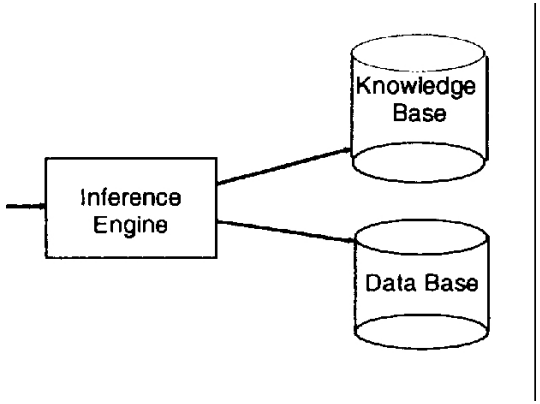
Knowledge : from Where ?

- AI needs much knowledge about our world
- Explicit-symbolic approach :
  Cyc = hand-crafted collection of rules and facts

Gigantic



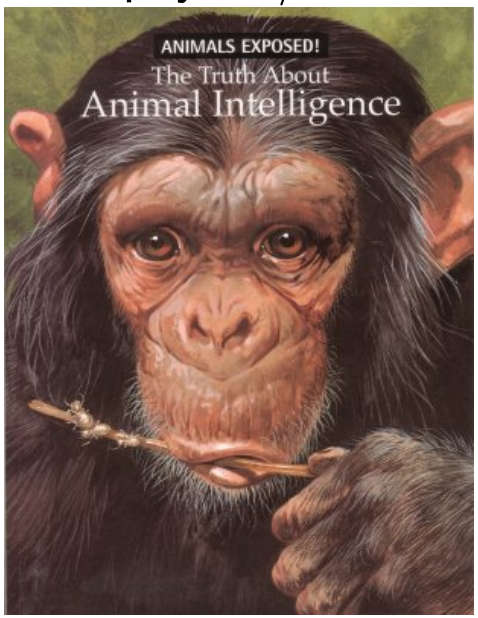
Incoherent/Incomplete



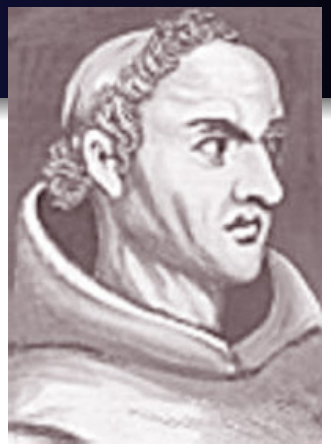
Not robust
not uncertainty-friendly

Learning the Knowledge ?

- Animals and humans : innate + learned
- Can learn many tasks that were not evolution-tuned !
- Bet on existence of some generic strategies/principles
- More exciting / greater payoff / worth exploring this path



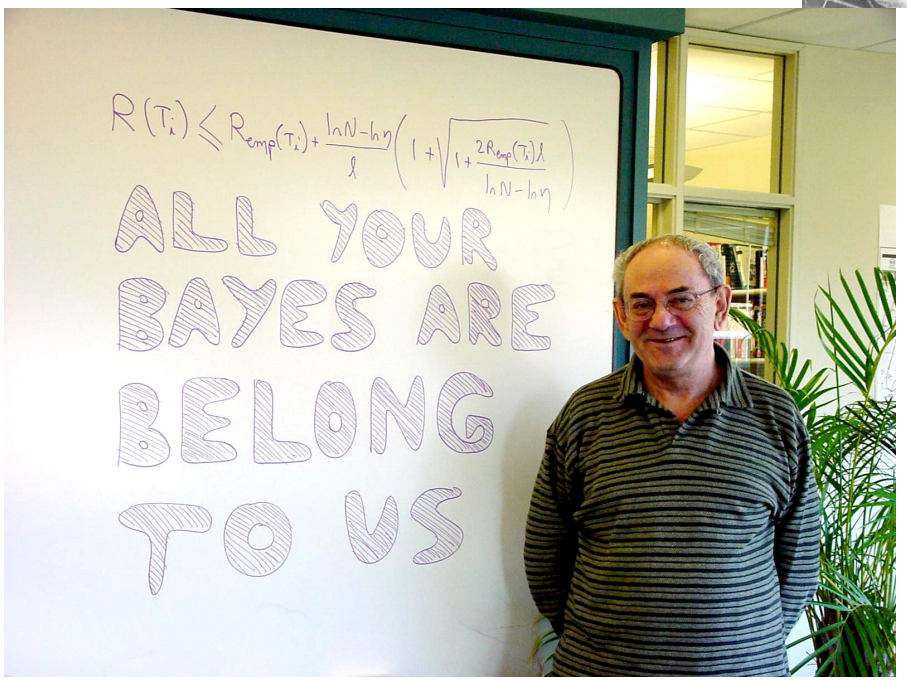
Compact Representation ⇒ Generalization



Extract the essence from the data ⇒ generalization

- Occam's Razor

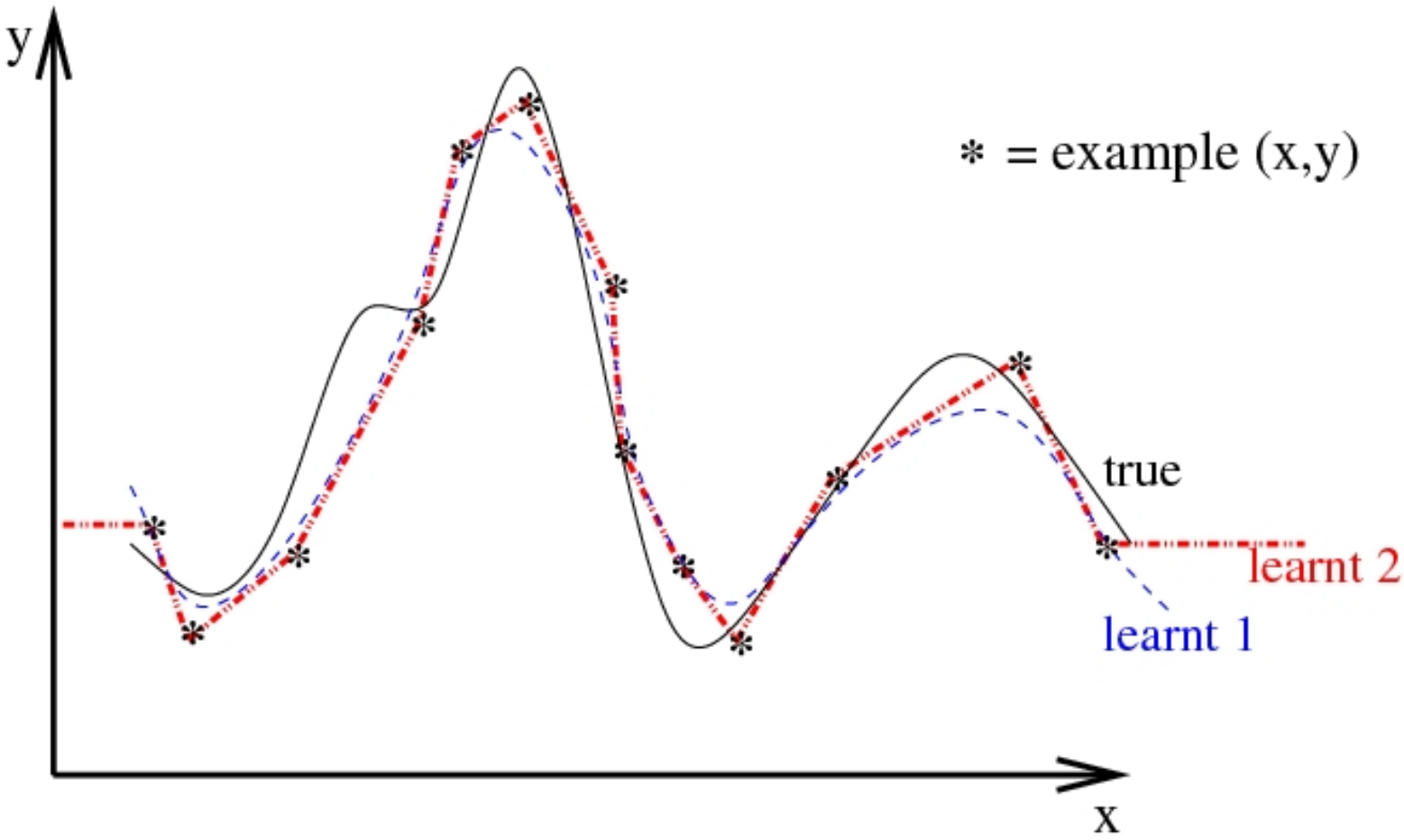
- Kolmogorov/Solomonoff 1964
- Vapnik 1972



$\min_f \frac{1}{n} \sum_{i=1}^{n} error(x_i, f(x_i)) \neq$
$\min_f E[error(x_i, f(x_i))]$

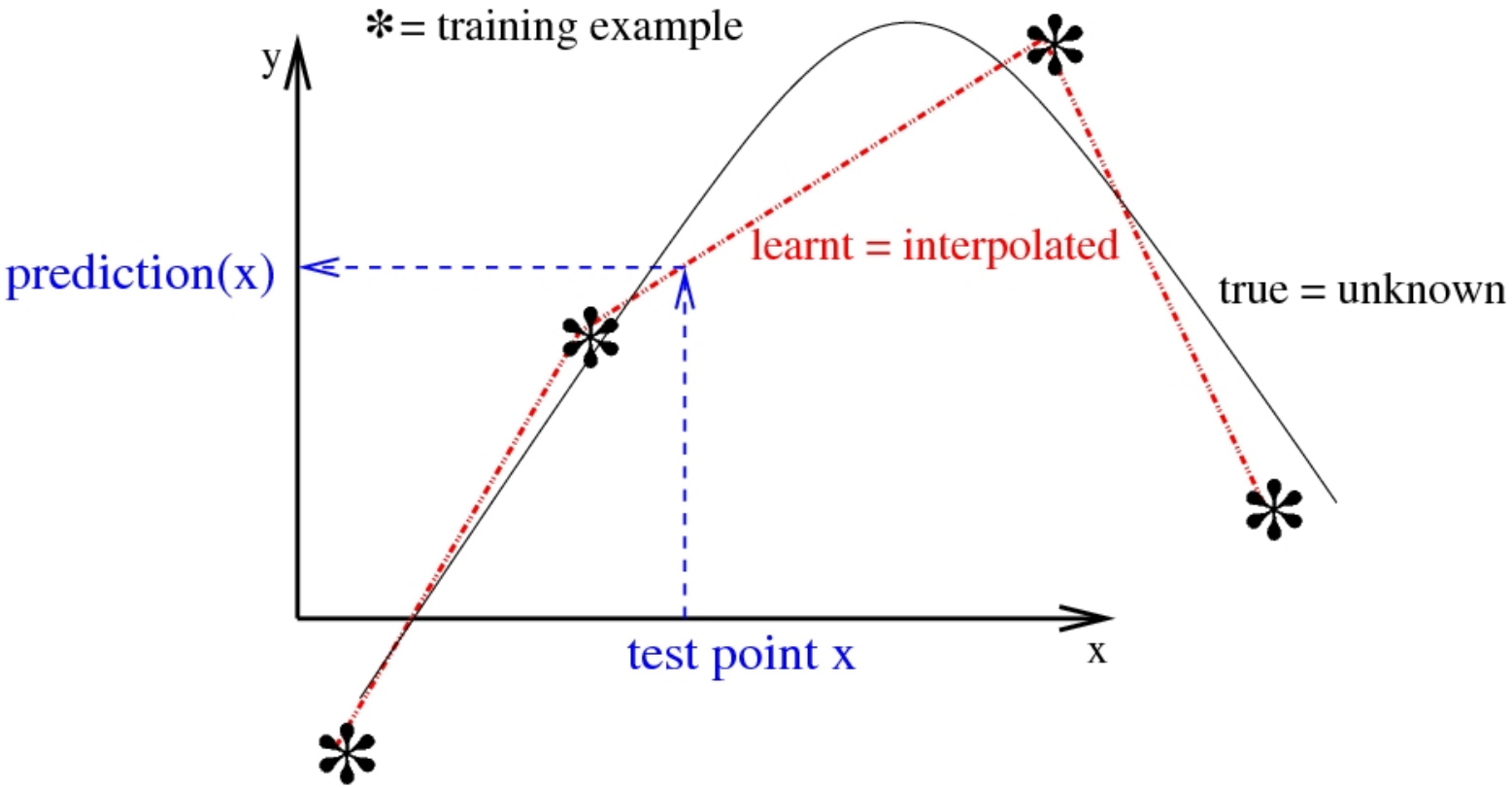
1D Generalization

Easy if the target function is smooth (few variations).



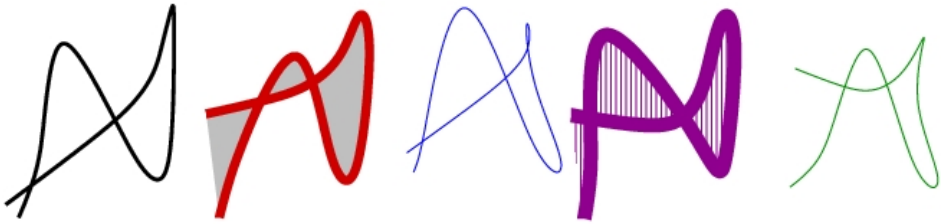
Generalizing Locally

Works well with a good representation : where notion of neighborhood is meaningful.



Where Local Generalization Fails

Training set : Klingon characters



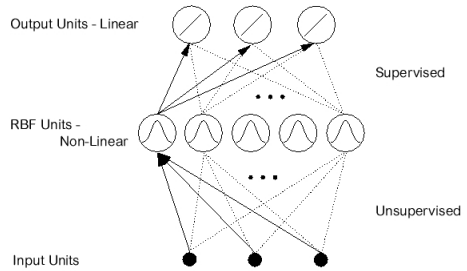
Test set :

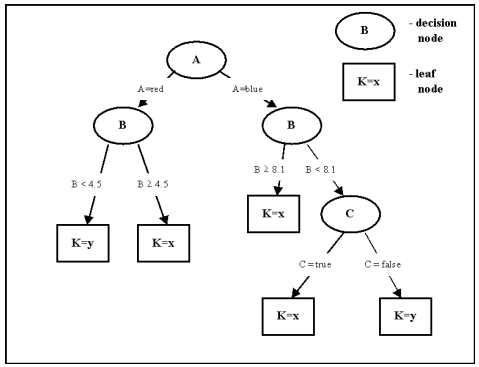
- Pixel-to-pixel Euclidean distance only works **very locally**
- Klingons work in a more abstract representation space

Local Generalization : negative results

We and others have shown negative results illustrating limitations of local generalization :

- (classical non-parametric)
- Local kernel methods :
  - SVMs
  - Gaussian Process
  - Graph-based semi- and un-supervised learning alg. (LLE, Isomap, etc.)
- Decision trees





All break data space in regions s.t.
degrees freedom $\propto$ # regions

Curse of Dimensionality

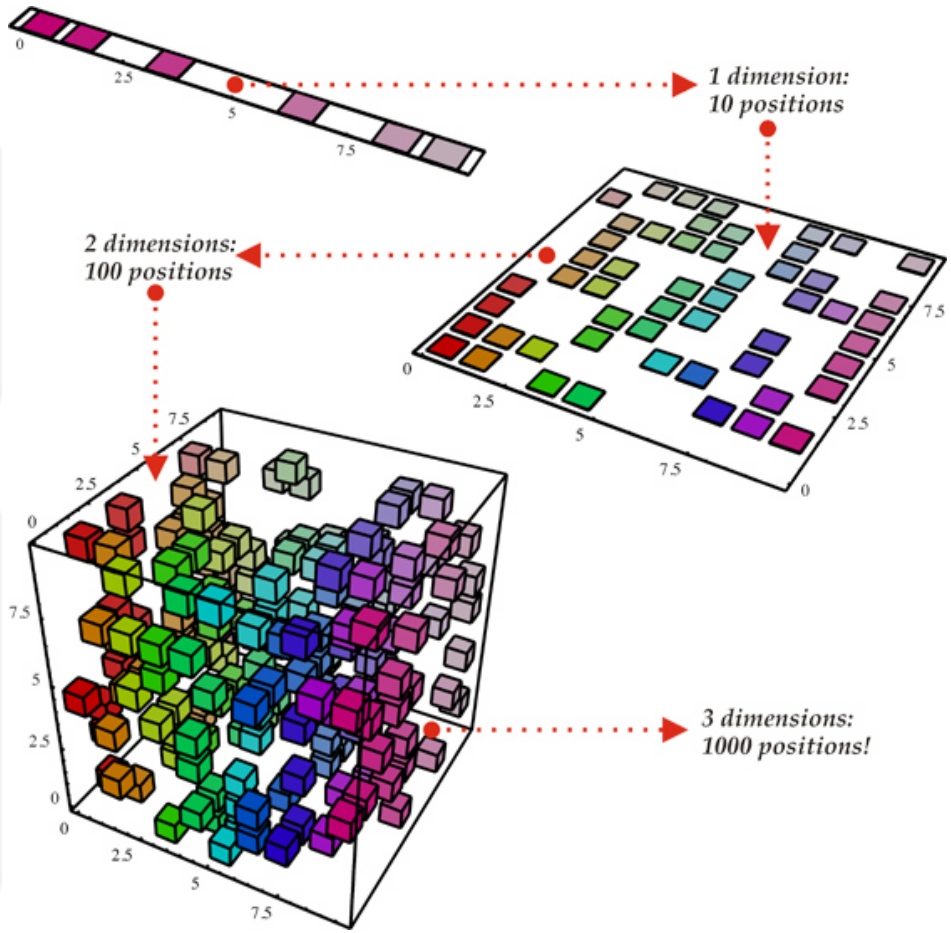
Basic result :

Theorem Sketch

Local learning algorithms cannot generalize to variations not covered by the training set.

Informal Corollary

Local learning algorithms can require a number of training examples exponential in the input dimension to obtain a given generalization error.

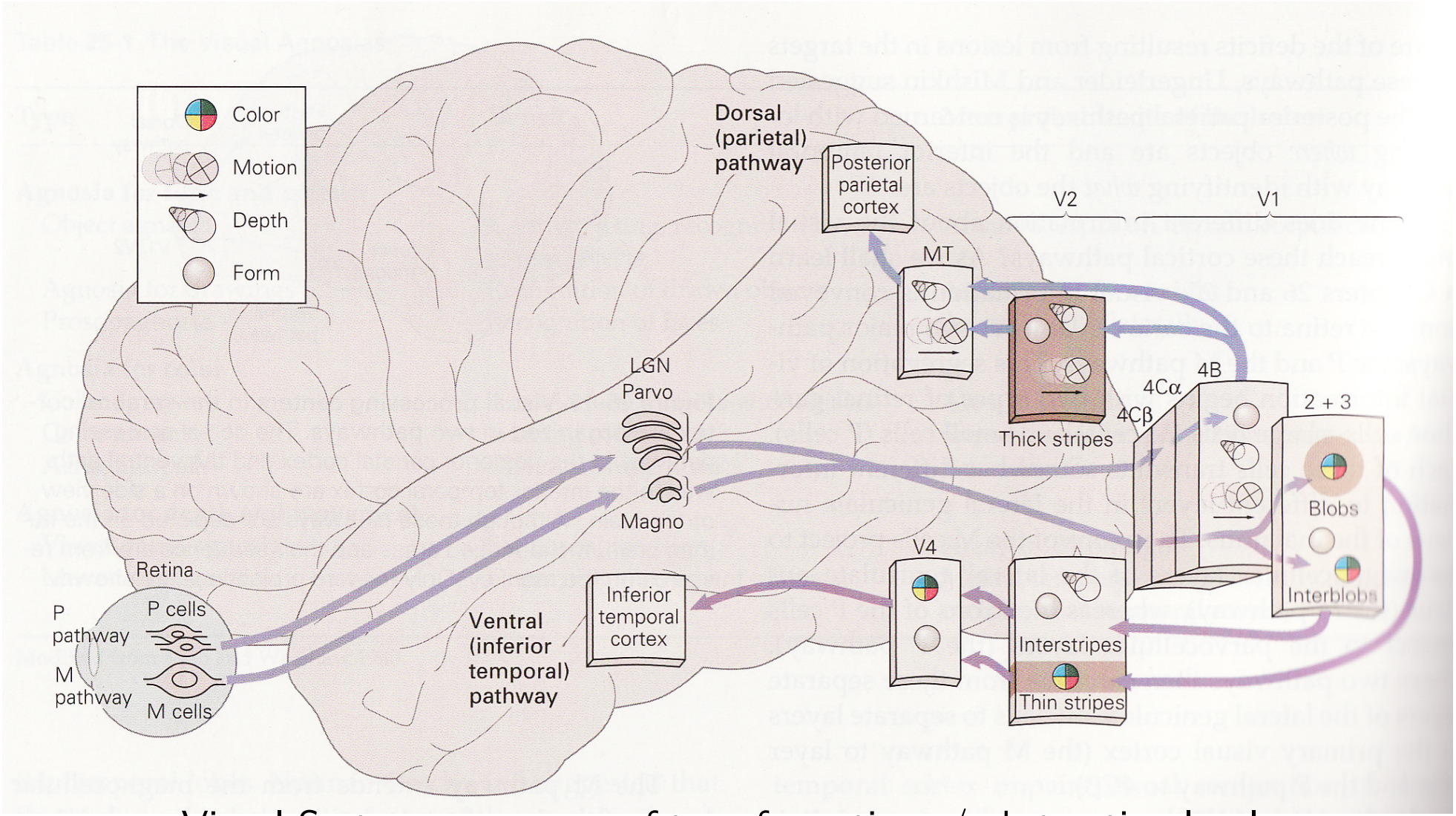


Curse of Dimensionality

Actual theoretical results (NIPS'2004, NIPS'2005, *Semi-Supervised Learning* book) are specialized :

- Gaussian kernel machines
  - Functions varying a lot along a straight line
  - Parity function
- Semi-supervised learning from neighborhood-graph
- Decision trees on highly varying functions
- Local kernel manifold learning from neighborhood-graph (such as kPCA, LLE, Isomap, ...)

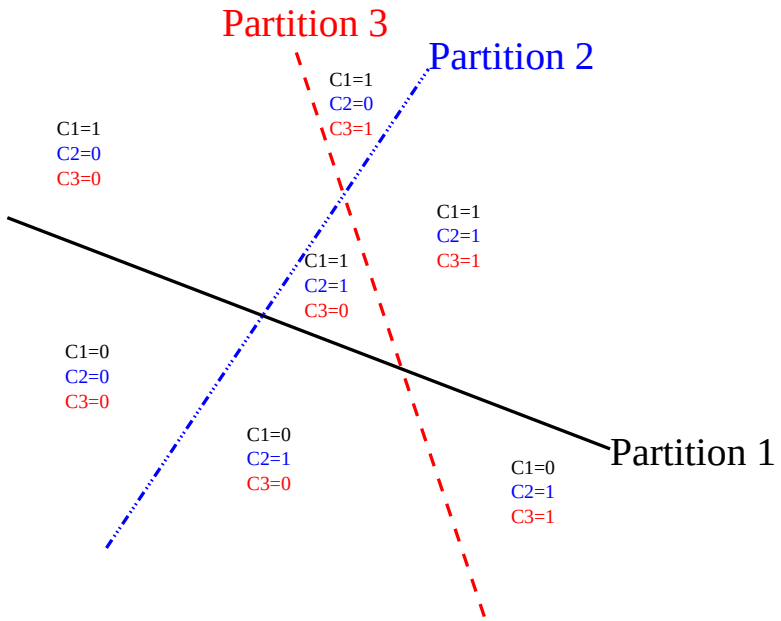
Primates Visual System



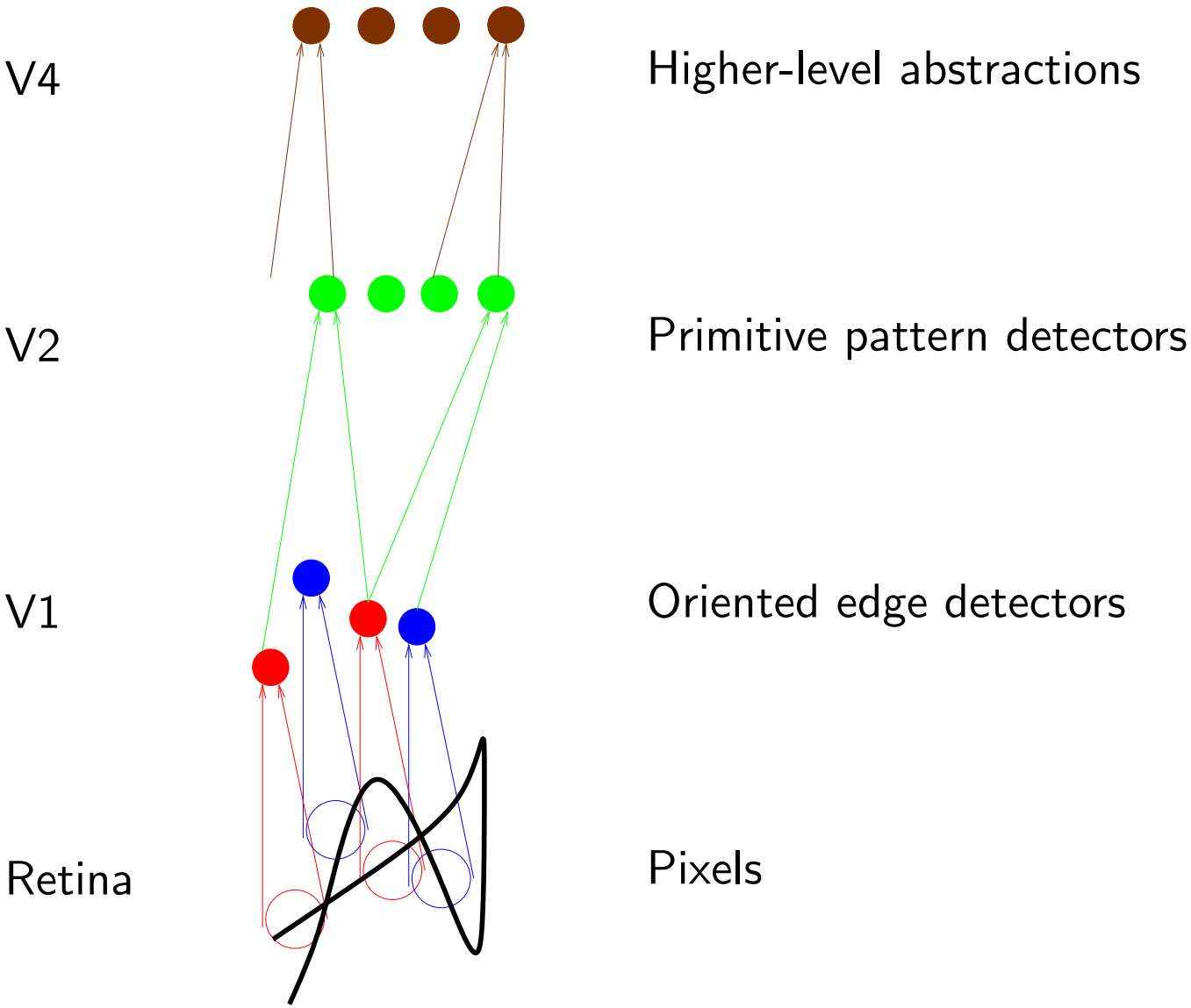
Visual System : sequence of transformations / abstraction levels

Strategy : Distributed Representations

- Distributed representation : each percept represented by the combination of many features
- Multiple levels of representation (like in the brain)
- **Exponential advantage**
- Missing in many current learning algorithms (most clustering, non-parametric, semi-supervised, kernels, mixture models, etc.)

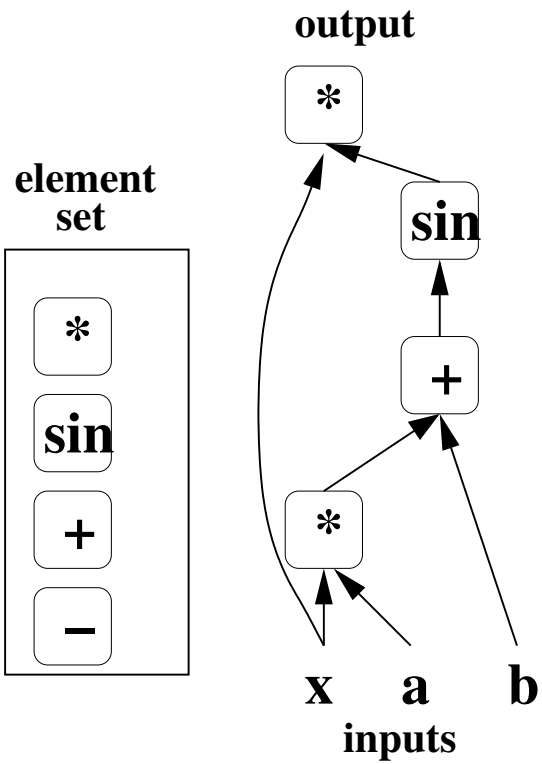


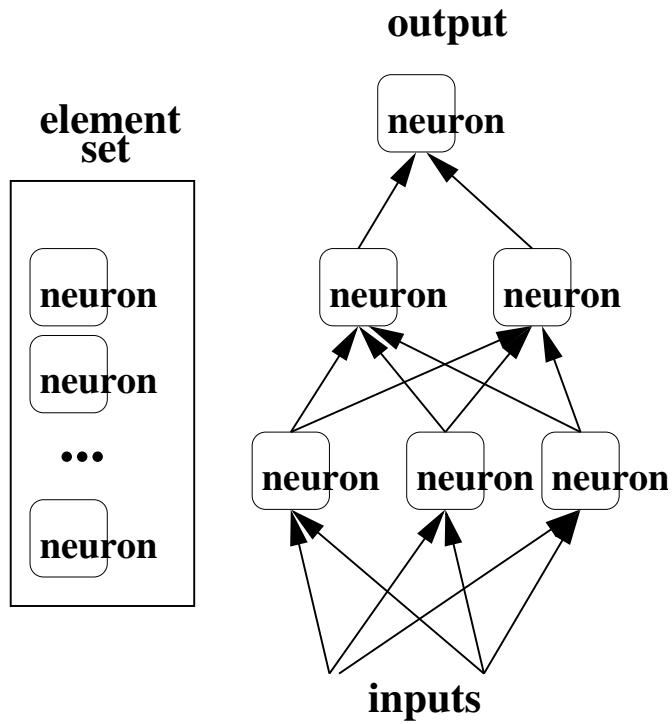
Exploiting Multiple Levels of Representation



Computation Graph and Depth

Each node = computation element (from a set)





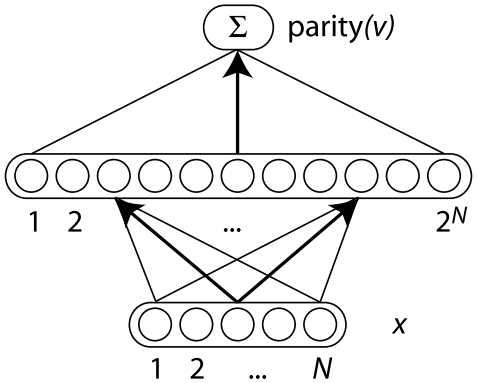
compute $x * \sin(a * x + b)$,
depth=4.

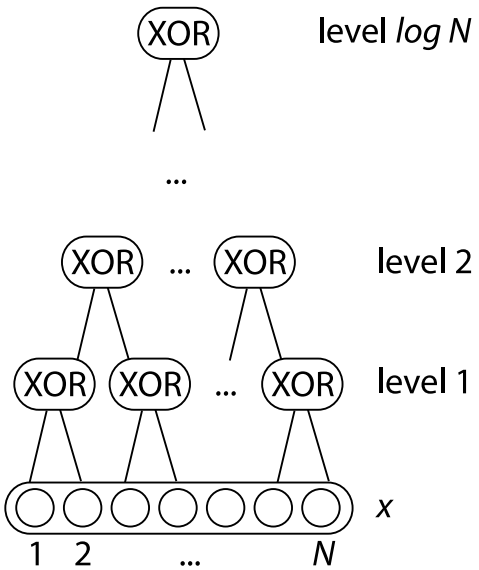
elements = artificial neurons $f(x) = \tanh(b + w'x)$.
Multi-layer neural net
depth=3.

Architecture Depth

Theorem Sketch

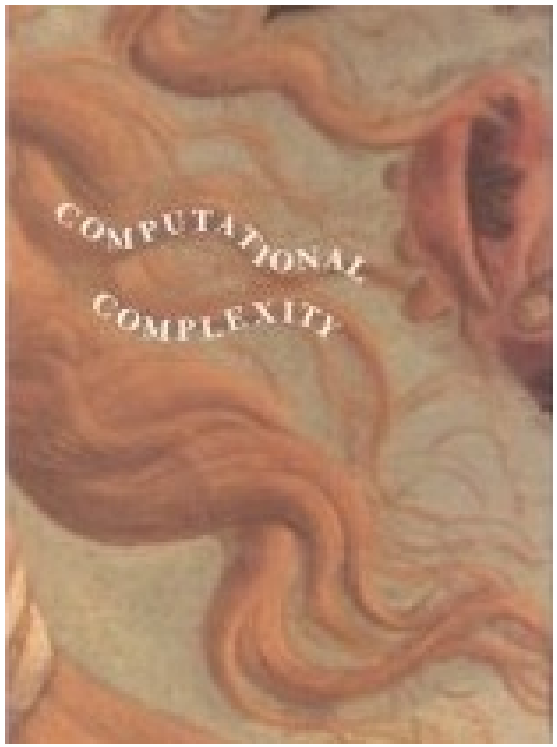
When a function can be compactly represented by a deep architecture, representing it with a shallow architecture can require a number of elements exponential in the input dimension.





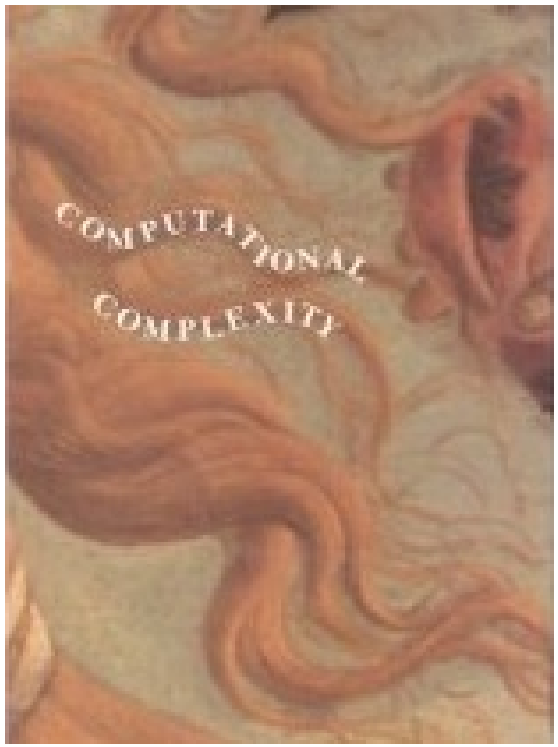
Insufficient Depth

- 2-level logic circuits on $\{0,1\}^n$
  - can represent any discrete function
  - most functions require $O(2^n)$ logic gates
  - $\exists$ functions computable efficiently with depth k, requiring $O(2^n)$ gates if depth $\leq k-1$.



Insufficient Depth

- **Similar results for circuits of formal neurons**
- $\exists$ "simple" functions requiring exponential size architectures with Gaussian kernel machines
- Most current/popular learning algorithms : **depth = 1 ou 2**, sometimes 3 (e.g. boosted trees, forests, 2-hidden layer MLPs)

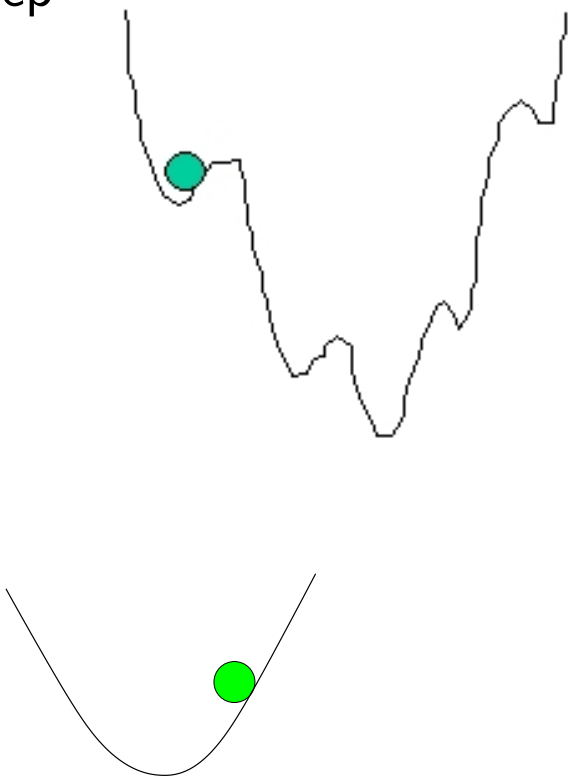


Insufficient Depth : Consequence

- Theoretical results (proved for some element sets)
  insufficient depth $\Rightarrow$
  very fat architecture
- $\Rightarrow$ very large number of parameters required
- $\Rightarrow$ very large number of examples required

Training Deep Architectures : the Challenge

- Two levels suffice to represent any function
- Up to 2006, failure of attempts to train deep architectures
- Why ? Non-convex optimisation and stochastic !
- Focus NIPS 1995-2005 : convex learning algorithms



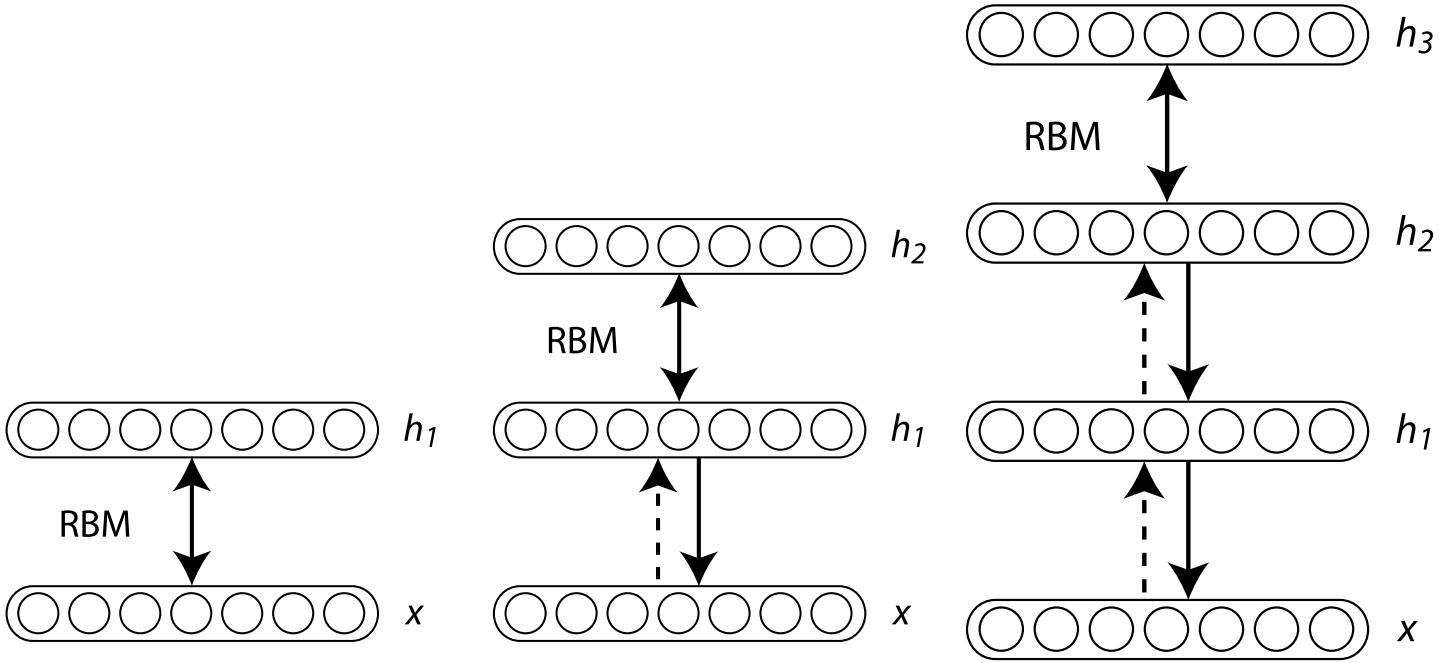
$\Rightarrow$ Let us face the challenge !

Strategy : multi-task semi-supervised learning

- Humans : most examples not semantically labeled
- Each example informs many tasks
- Representations shared among tasks
- Capture common factors of variations to generalize easily to new tasks (even with 0 examples !)
- $\lim_{\#\ tasks \to \infty}$ = (un+semi)-supervised

Strategy : one level of abstraction at a time

- Humans learn simple things first :
  first levels of visual system converge in critical periods
- Percept representation = abstraction of the percept
- Learn a first level of representation, then a second built on the first, etc.



Deep Belief Network (DBN)

2006 : Breakthrough !

- FIRST : successful training of **deep architectures !**
  Hinton et al Neural Comp. 2006, followed by Bengio et al, and Ranzato et al at NIPS'2006
- One trains one layer after the other of an MLP
- Unsupervised learning in each layer of initial representation
- Continue training an ordinary but deep MLP near a better minimum

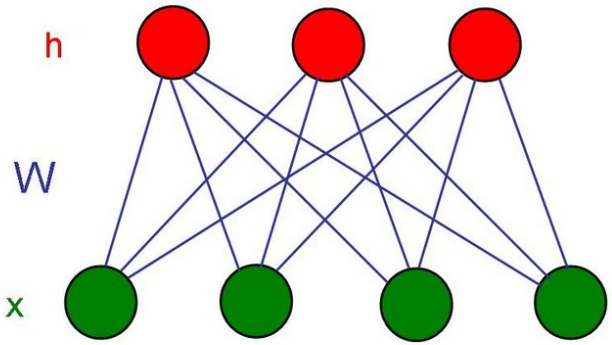
Individual Layer : RBMs and auto-encoders

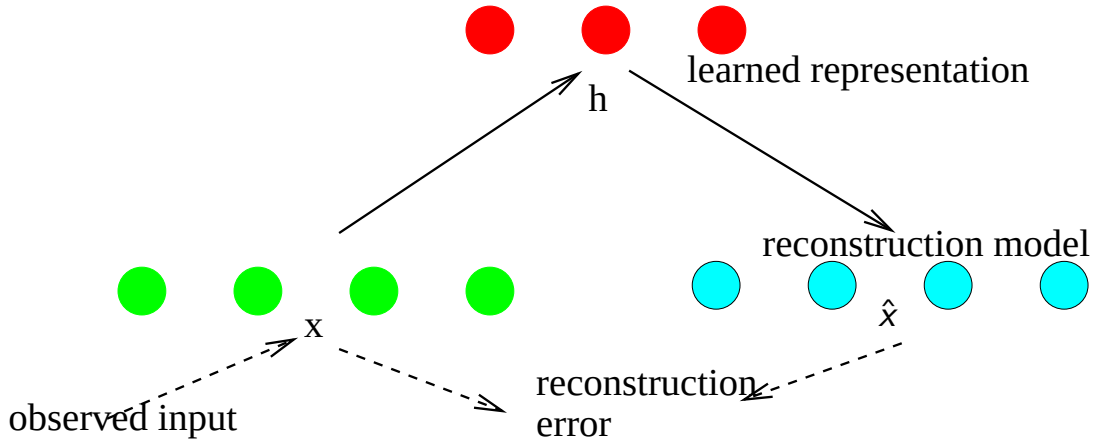
Restricted Boltzmann Machine :

$$P(x) = \sum_h P(x,h) \propto \sum_h e^{h' W x}$$

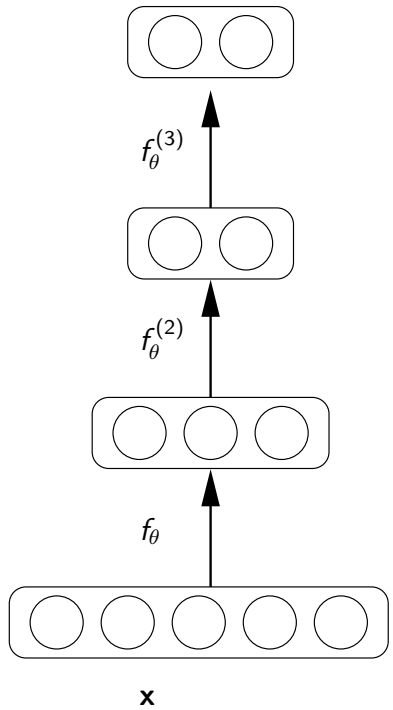
Allows efficient inference
of causes $h = (h_1, \ldots, h_n)$

Auto-encoder : Looks for a compact representation $h(x)$ of x
$x \Rightarrow h(x) \Rightarrow \hat{x}(h(x))$
$\Rightarrow$ reconstruction error$(x, \hat{x})$
Learn functions
$h(\cdot)$ and $\hat{x}(\cdot)$.



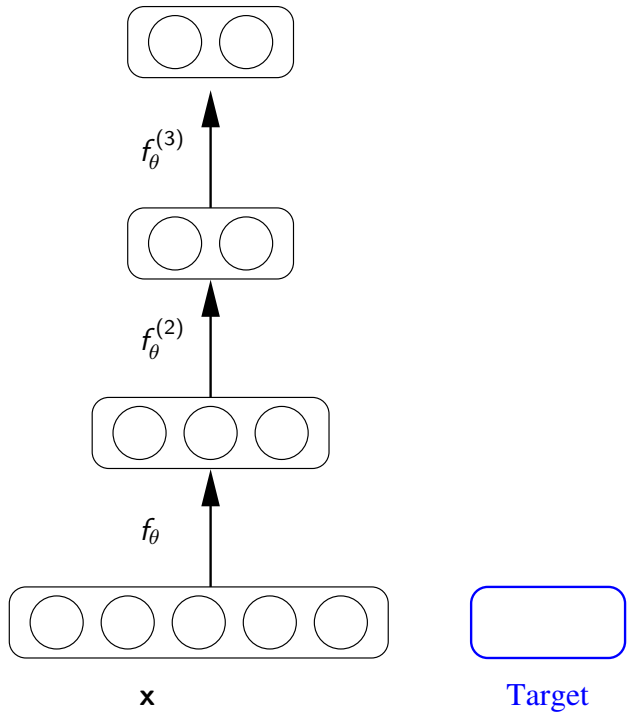


- Initial deep hierarchical mapping is learnt in an unsupervised way.
- $\longrightarrow$ initialization for a supervised task.
- Output layer gets added.
- Global fine tuning by gradient descent on supervised criterion.



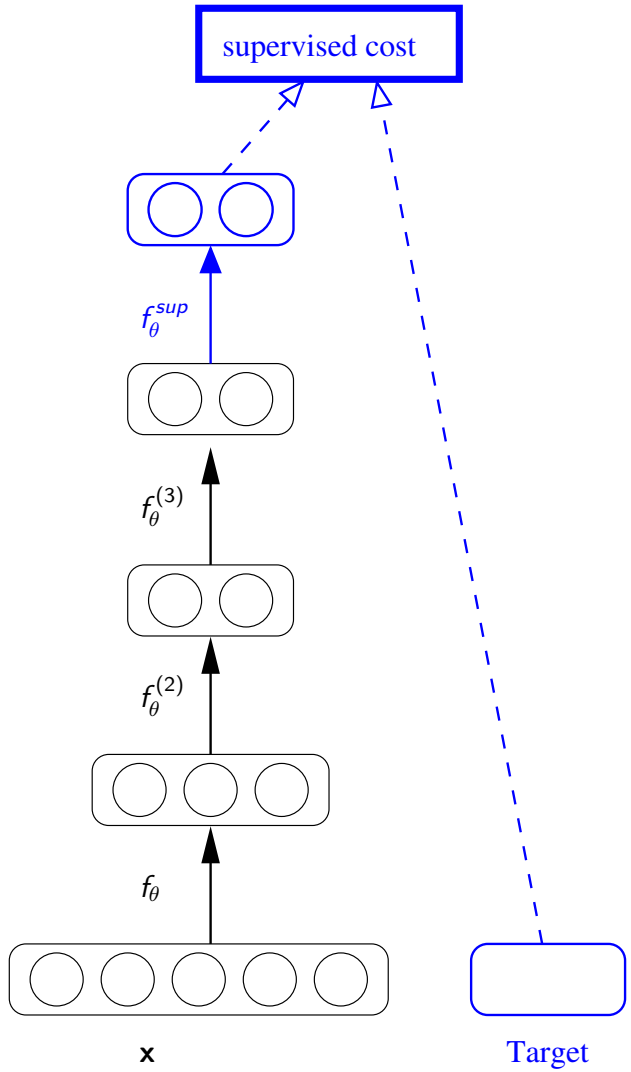
Learning deep networks
Supervised fine-tuning

- Initial deep hierarchical mapping is learnt in an unsupervised way.
- $\longrightarrow$ initialization for a supervised task.
- Output layer gets added.
- Global fine tuning by gradient descent on supervised criterion.



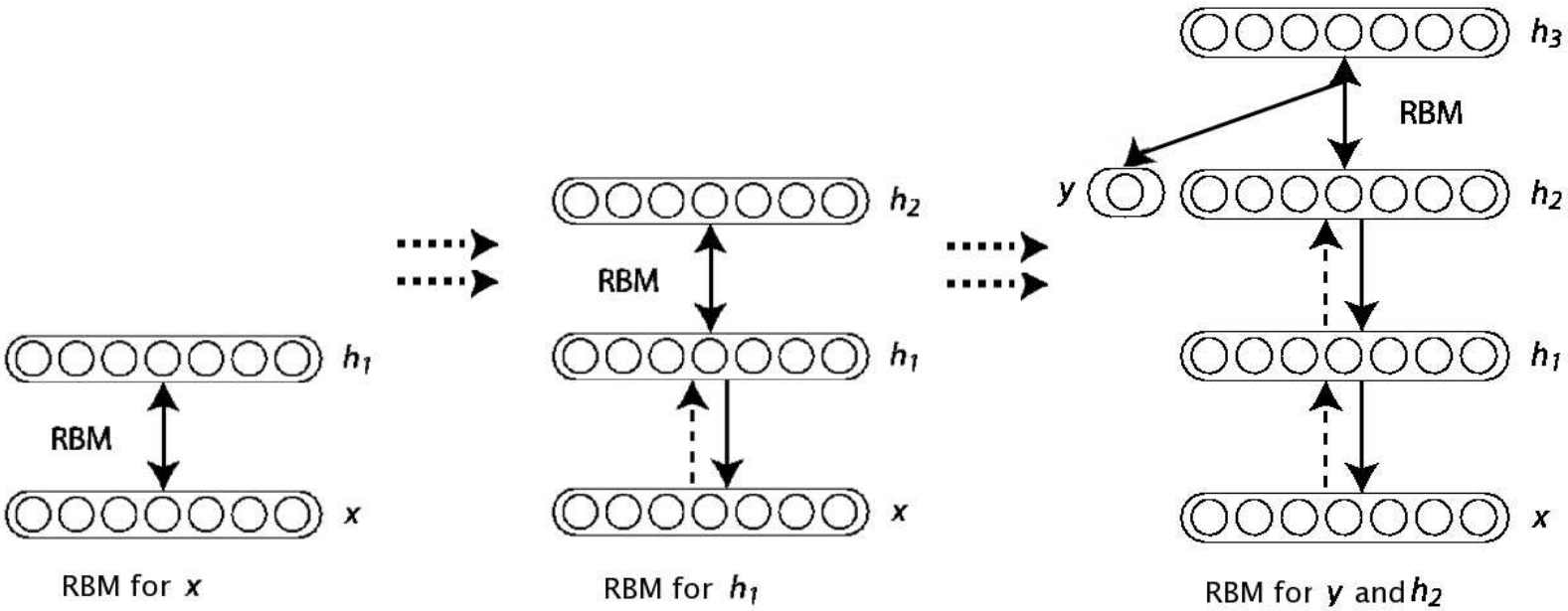
Learning deep networks
Supervised fine-tuning

- Initial deep hierarchical mapping is learnt in an unsupervised way.
- $\longrightarrow$ initialization for a supervised task.
- Output layer gets added.
- Global fine tuning by gradient descent on supervised criterion.



Deep Belief Nets

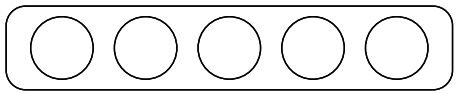
- Hinton et al. (2006) introduced the Deep Belief Network (DBN), a deep probabilistic/generative neural network
- The training procedure is first **layer-wise greedy** and **unsupervised** (initialization).



- Then the model is converted into a conditional predictor and **fine-tuned**

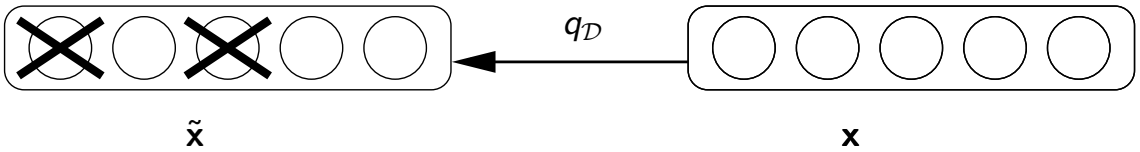
$$\min_{\theta} \; -\frac{1}{n} \sum_{i=t}^{n} \log \hat{p}(y_t | x_t, \theta)$$

Denoising Auto-Encoders



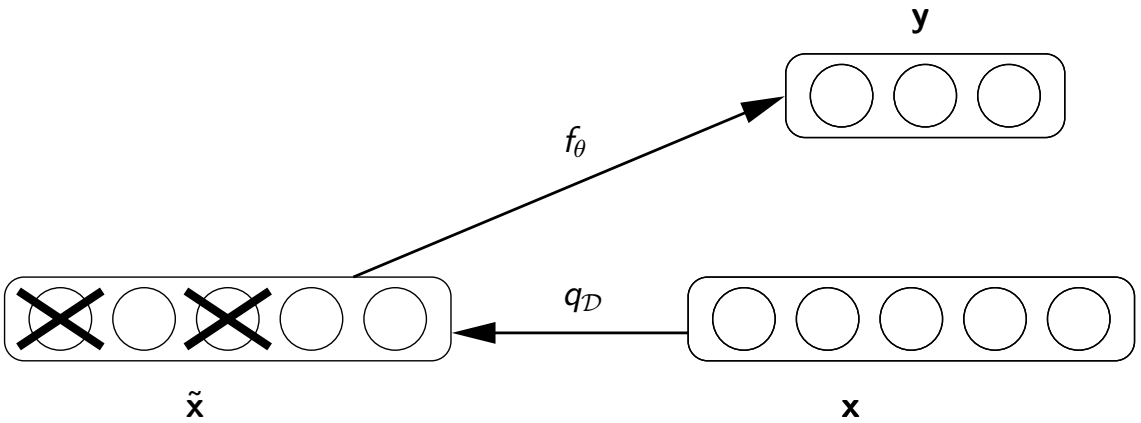
- Clean input $\mathbf{x} \in [0,1]^d$ is partially destroyed, yielding corrupted input : $\tilde{\mathbf{x}} \sim q_{\mathcal{D}}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to hidden representation $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From $\mathbf{y}$ we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the cross-entropy "reconstruction error"
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders



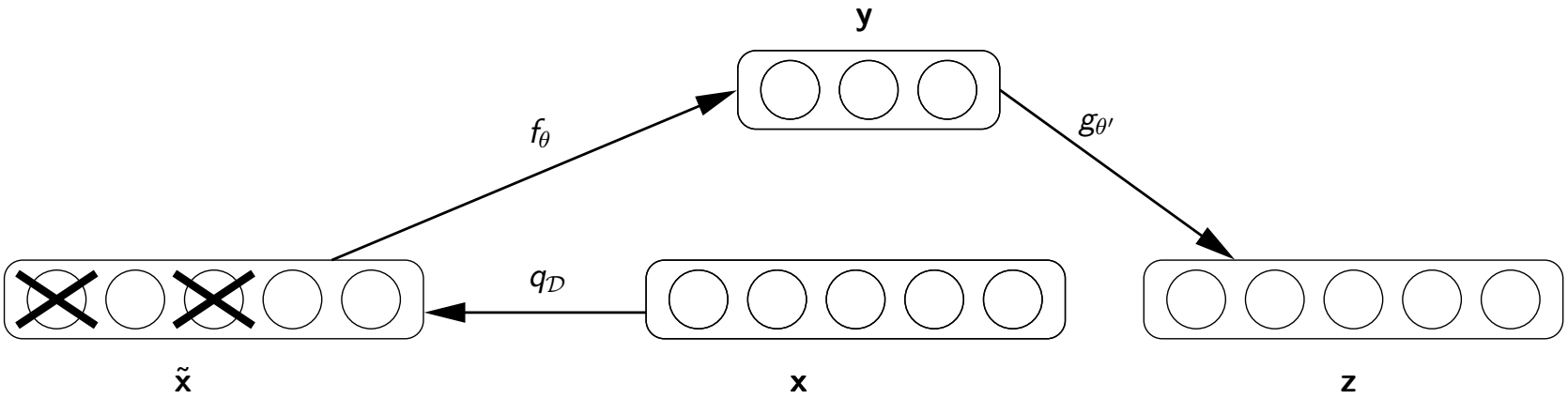
- Clean input $\mathbf{x} \in [0,1]^d$ is partially destroyed, yielding corrupted input : $\tilde{\mathbf{x}} \sim q_{\mathcal{D}}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to hidden representation $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From $\mathbf{y}$ we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the cross-entropy "reconstruction error"
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders



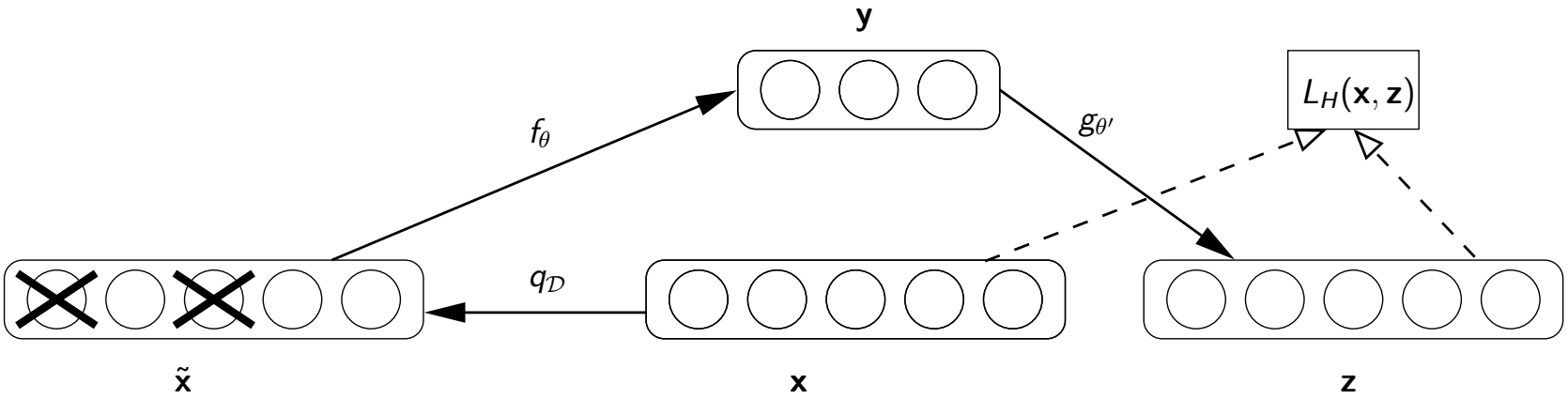
- Clean input $\mathbf{x} \in [0,1]^d$ is partially destroyed, yielding corrupted input : $\tilde{\mathbf{x}} \sim q_\mathcal{D}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to hidden representation $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From $\mathbf{y}$ we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the cross-entropy "reconstruction error"
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders



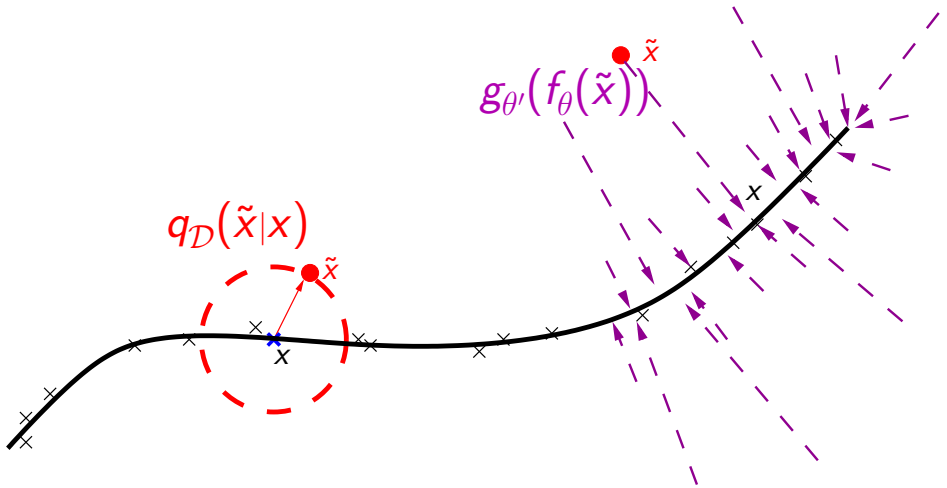
- Clean input $\mathbf{x} \in [0,1]^d$ is partially destroyed, yielding corrupted input : $\tilde{\mathbf{x}} \sim q_{\mathcal{D}}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to hidden representation $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From $\mathbf{y}$ we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the cross-entropy "reconstruction error"
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders



- Clean input $\mathbf{x} \in [0,1]^d$ is partially destroyed, yielding corrupted input : $\tilde{\mathbf{x}} \sim q_\mathcal{D}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to hidden representation $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From $\mathbf{y}$ we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the cross-entropy "reconstruction error"
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Manifold Learning Perspective

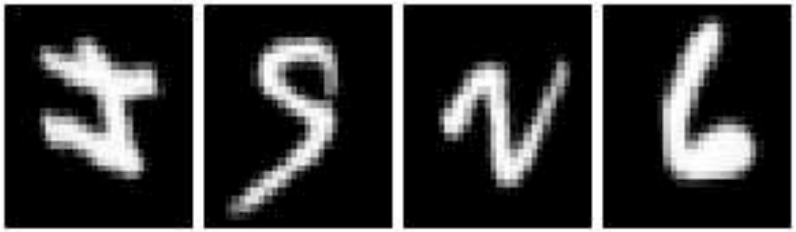


Denoising autoencoder can be seen as a way to learn a manifold :

- Suppose training data ($\times$) concentrate near a low-dimensional manifold.
- Corrupted examples ($\bullet$) are obtained by applying corruption process $q_{\mathcal{D}}(\widetilde{X}|X)$ and will lie farther from the manifold.
- The model learns with $p(X|\widetilde{X})$ to "project them back" onto the manifold.
- Intermediate representation Y can be interpreted as a coordinate system for points on the manifold.

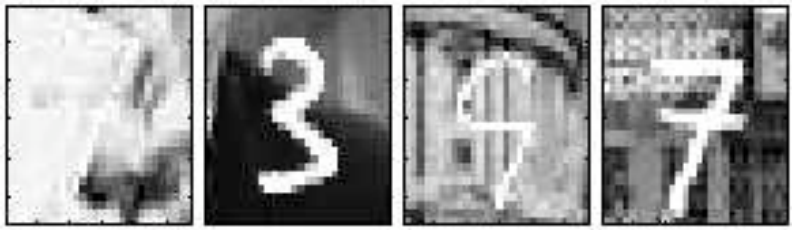
Benchmark problems
Variations on MNIST digit classification

basic : subset of original MNIST digits : 10 000 training samples, 2 000 validation samples, 50 000 test samples.

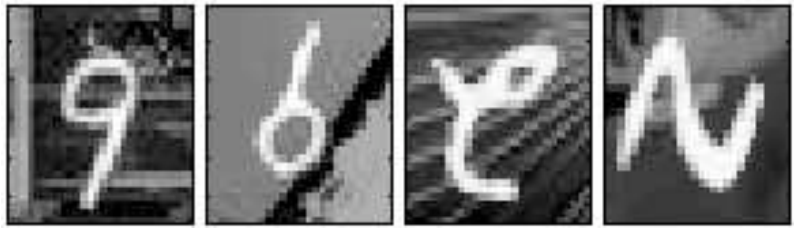


(a) **rot :** applied random rotation (angle between 0 and 2π radians)

(b) **bg-rand :** background made of random pixels (value in $0 \ldots 255$)



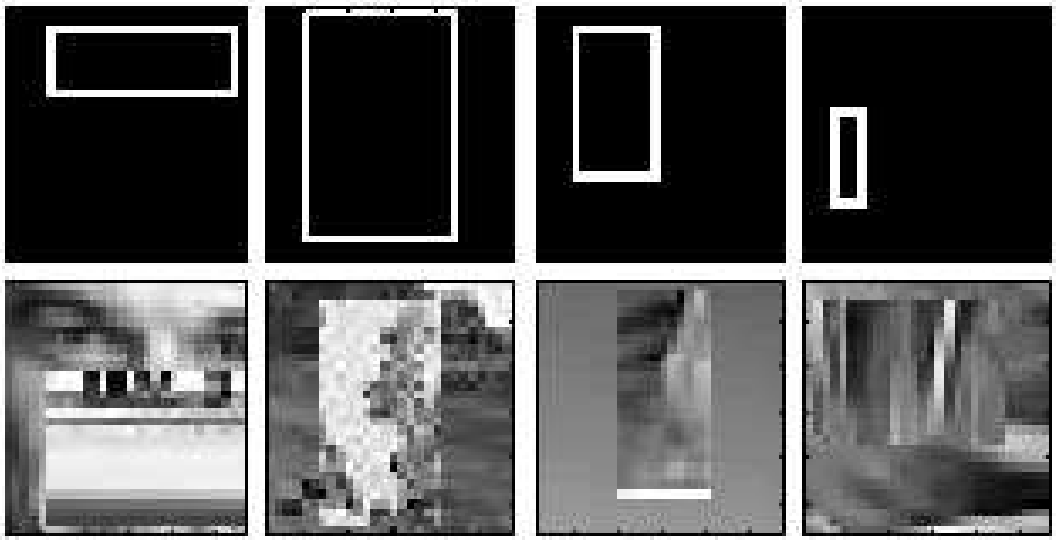
(c) **bg-img :** background is random patch from one of 20 images



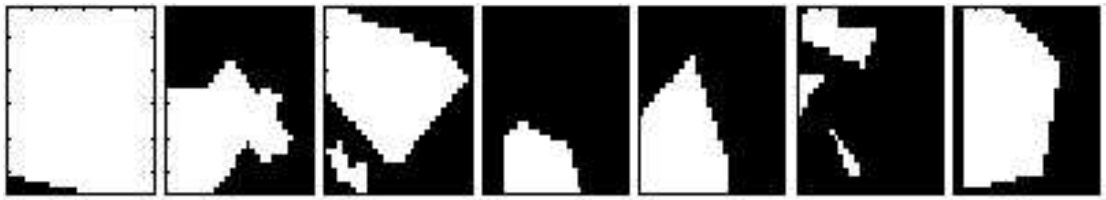
(d) **rot-bg-img :** combination of rotation and background image

Benchmark problems
Shape discrimination

- **rect :** discriminate between tall and wide rectangles on black background.



- **rect-img** : borderless rectangle filled with random image patch. Background is a different image patch.
- **convex :** discriminate between convex and non-convex shapes.



Performance comparison

Results

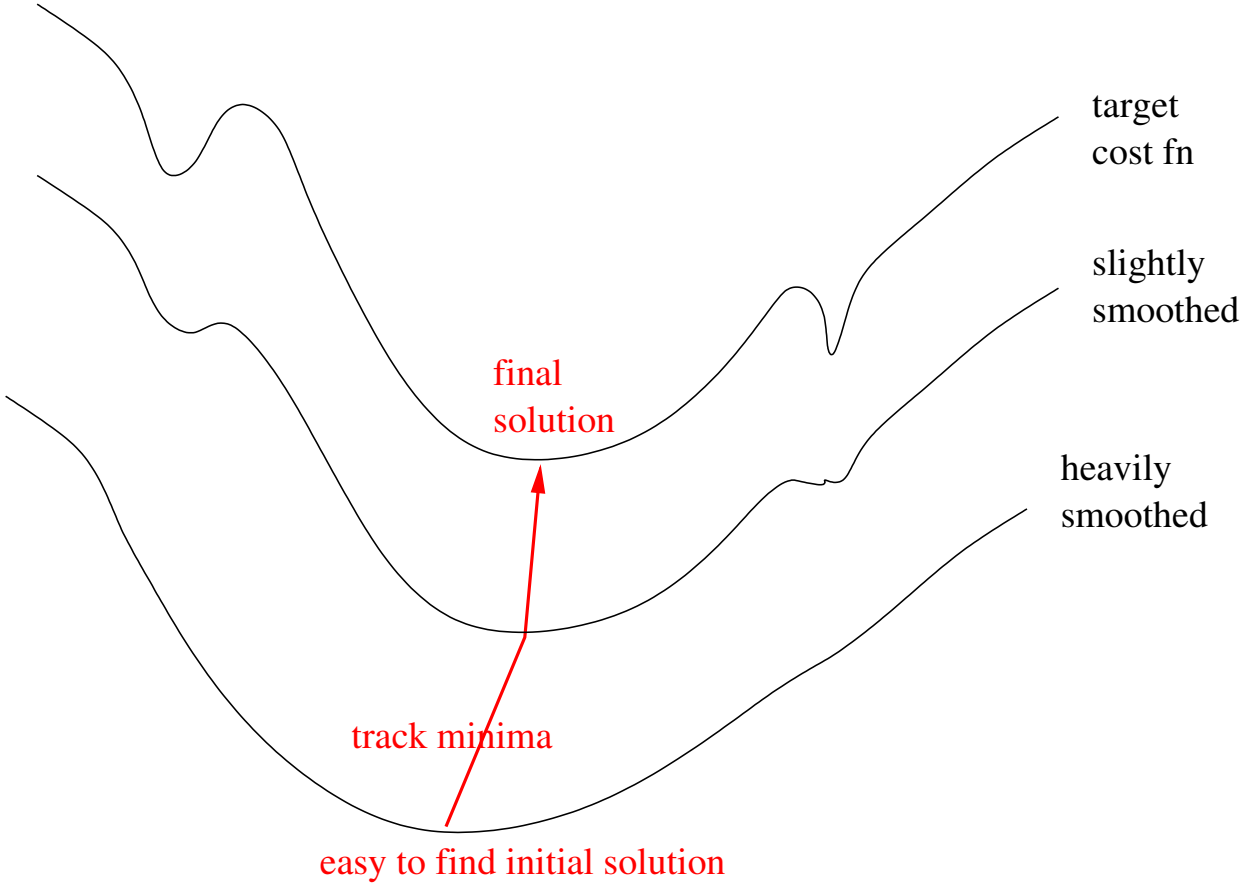
Dataset	SVM_{rbf}	DBN-3	SAA-3	SdA-3 (ν)
basic	$3.03_{\pm 0.15}$	$3.11_{\pm 0.15}$	$3.46_{\pm 0.16}$	$2.80_{\pm 0.14}$ (10%)
rot	$11.11_{\pm 0.28}$	$10.30_{\pm 0.27}$	$10.30_{\pm 0.27}$	$10.29_{\pm 0.27}$ (10%)
bg-rand	$14.58_{\pm 0.31}$	$6.73_{\pm 0.22}$	$11.28_{\pm 0.28}$	$10.38_{\pm 0.27}$ (40%)
bg-img	$22.61_{\pm 0.37}$	$16.31_{\pm 0.32}$	$23.00_{\pm 0.37}$	$16.68_{\pm 0.33}$ (25%)
rot-bg-img	$55.18_{\pm 0.44}$	$47.39_{\pm 0.44}$	$51.93_{\pm 0.44}$	$44.49_{\pm 0.44}$ (25%)
rect	$2.15_{\pm 0.13}$	$2.60_{\pm 0.14}$	$2.41_{\pm 0.13}$	$1.99_{\pm 0.12}$ (10%)
rect-img	$24.04_{\pm 0.37}$	$22.50_{\pm 0.37}$	$24.05_{\pm 0.37}$	$21.59_{\pm 0.36}$ (25%)
convex	$19.13_{\pm 0.34}$	$18.63_{\pm 0.34}$	$18.41_{\pm 0.34}$	$19.06_{\pm 0.34}$ (10%)

Some Results 2006-2008

- Deep architectures superior on MNIST (NIPS'2006)
- Greater advantage on more complex tasks (ICML'2007)
- RBMs slightly better than ordinary auto-encoders
- Denoising auto-encoders $\geq$ RBMs, and more flexible
- Applications in NLP, vision, MOCAP, collaborative filtering
- Optimization sometimes still deficient, challenges ahead

Understanding Why it Works

Continuation methods



Several Strategies are Continuations

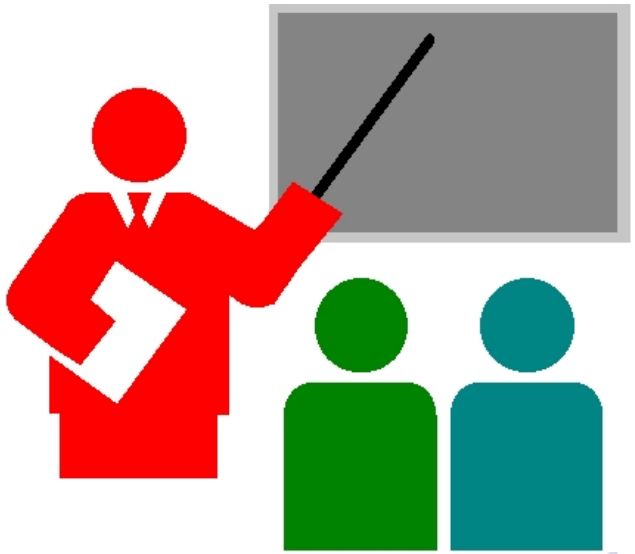
- Older : stochastic gradient from small parameters
- Breakthrough : greedy layer-wise construction
- New : gradually bring in more difficult examples



Curriculum Strategy

Start with simpler, easier examples, and gradually introduce more of the more complicated ones as the learner is ready to learn them.

Design the sequence of tasks / datasets to guide learning/optimization.

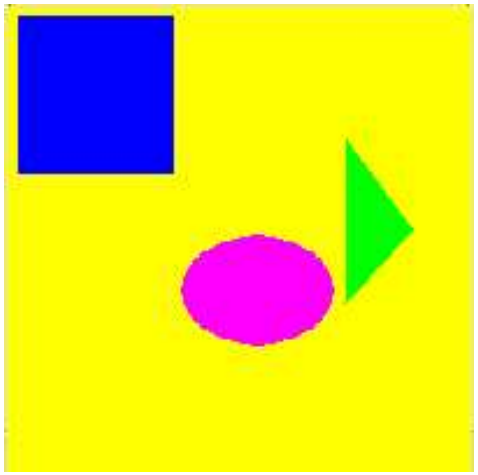


Strategy : Society = Parallel Optimisation

- Each human = potential solution
- Better solutions spread through language
- Similar to genetic evolution : parallel search + recombination
- R. Dawkins' *Memes*
- Simulations support this hypothesis
- AI : take avantage of human culture

Baby AI Project

Combine many strategies, to obtain a baby AI that masters the semantics of a simple visual + linguistic universe



Subject	**Question**	**Answer**
Color	There is a small triangle. What color is it ?	Green
Shape	What is the shape of the green object ?	Triangle
Location	Is the blue square at the top or at the bottom ?	At the top
Size	There is a triangle on the right.	
	Is it rather small or big ?	Small
Size (relative)	Is the square smaller or bigger than the triangle ?	Bigger

Conclusions and work in progress

- AI $\Rightarrow$ learning $\Rightarrow$ generalize
  $\Rightarrow$ generalize non-locally $\Rightarrow$ learn distributed representations
  $\Rightarrow$ deep architectures $\Rightarrow$ optimisation challenge
- **Breakthrough in 2006**
- Biological inspiration : humans' strategies to optimize learning of world model
  - multi-task, unsupervised, semi-supervised
  - multiple levels of distributed representation
  - learn lower levels first
  - curriculum / education
  - collective parallel search
- Ongoing work : denoising auto-encoders, work as well or better than DBNs
- Collobert & Weston : learning representations by layer-wise manifold learning
- *Patience : see the long term...*