Yoshua Bengio & **Olivier Delalleau**
Statistical Learning Algorithms Canada Research Chair, U. Montreal

ALT / DS 2011
October 5th, 2011, Espoo, Finland

ON THE EXPRESSIVE POWER OF DEEP ARCHITECTURES

From AI to Learning

- Intelligence requires knowledge → decisions
- Knowledge can be implicit
- Explicitly providing knowledge failed (expert systems)
  - Verbally expressed knowledge is incomplete
  - And lacks the required expression of uncertainty
- Learning captures knowledge from data
  - Can capture what is needed (completeness)
  - Can be actionable (learn to predict & act)
  - Can handle uncertainty (probabilistic models)

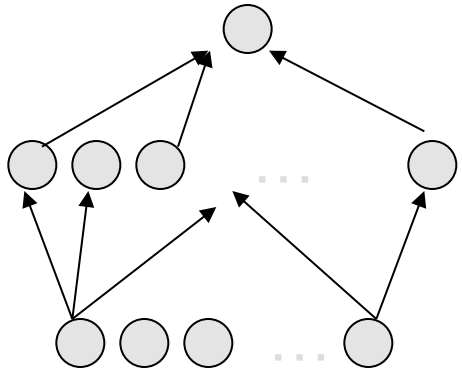
Learning to Generalize. How?

- Capturing dependencies between random variables
- Spreading out the probability mass from the empirical distribution. Where???
- Discovering underlying abstractions / explanatory factors

Onto Deep Learning

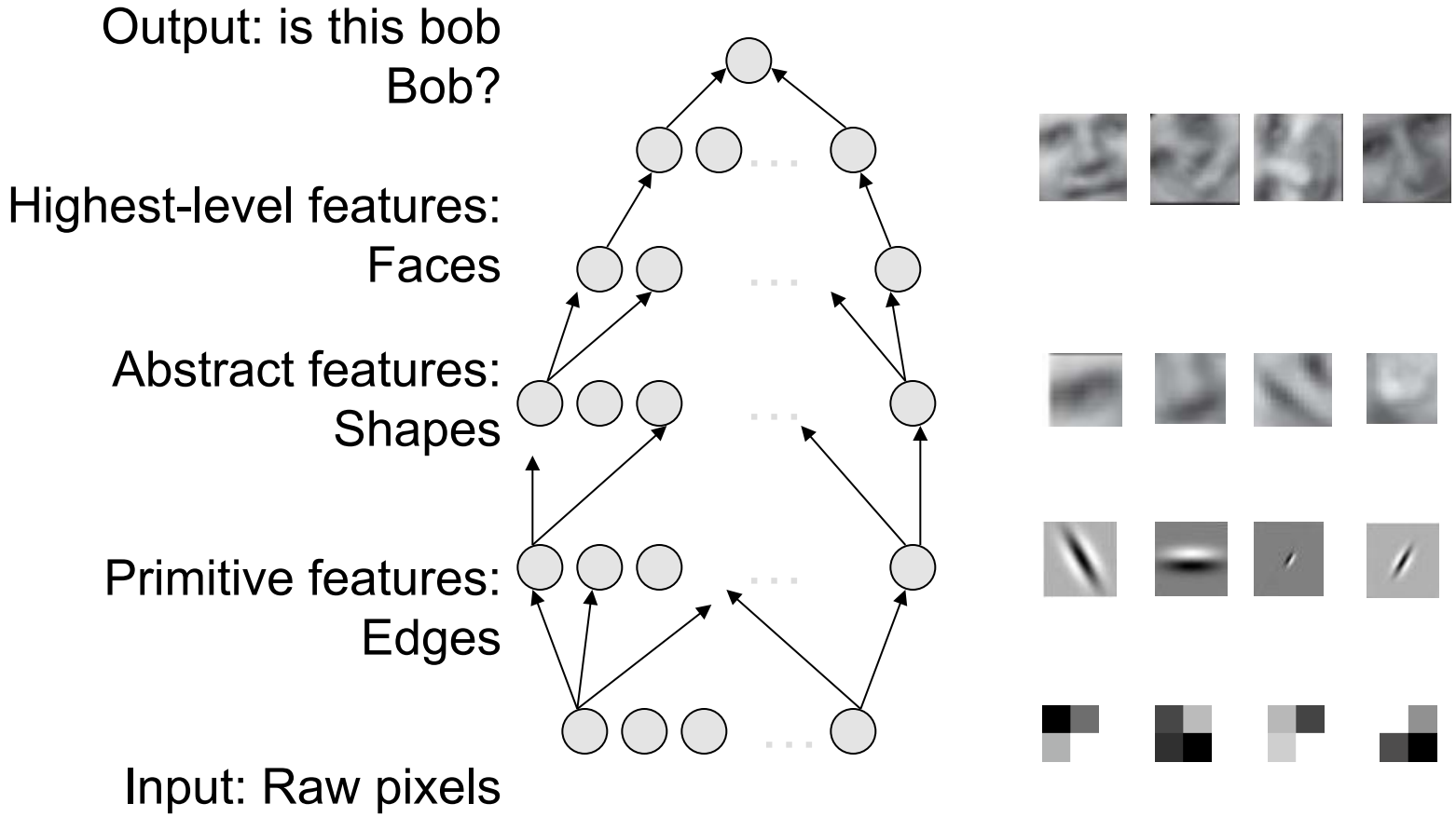
- Real-world distributions have convoluted unknown structure, not all captured by the principle of **local generalization**
- **We want weak priors that are stronger than the usual smoothness prior**
- Deep Learning: a way to address this by the discovery of multiple levels of representation capturing the underlying factors of variation

Shallow learning architecture



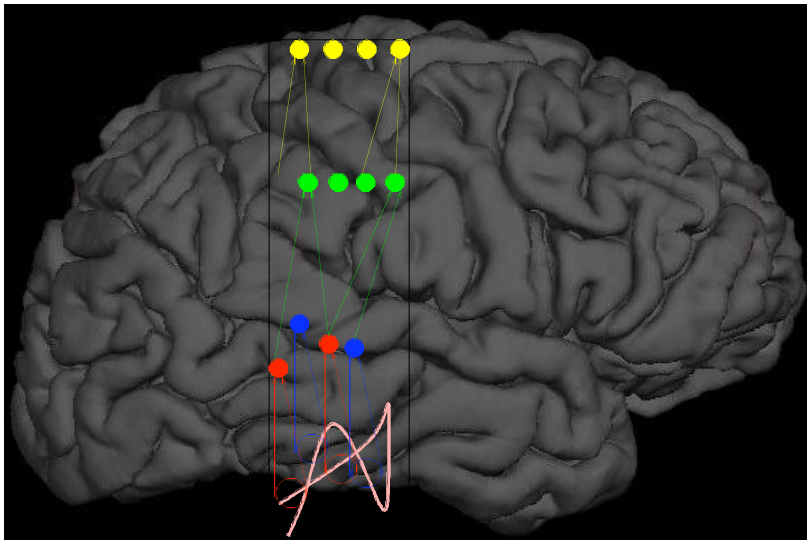
1-layer NNet, SVM, GP predictor, decision tree, boosted stumps, etc.

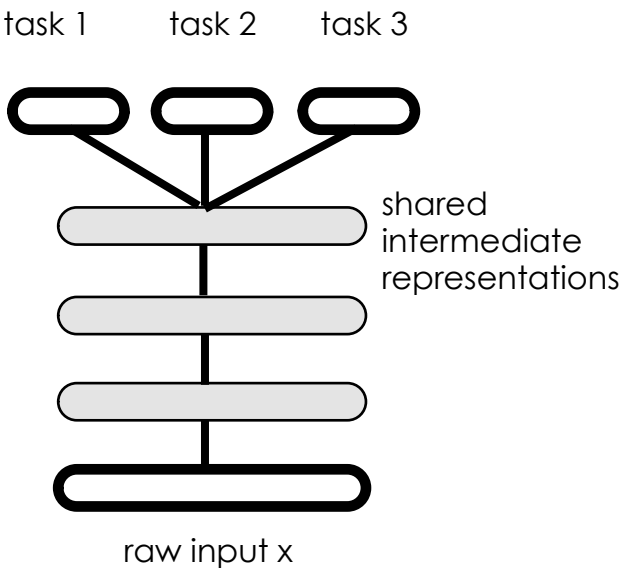
Deep learning architecture



Deep Motivations

- Brains have a deep architecture
- Cortex seems to have a generic learning algorithm
- Humans' ideas composed from simpler ones
- Insufficient depth can be exponentially inefficient
- Distributed (possibly sparse) representations necessary for non-local generalization, exponentially more efficient than 1-of-N enumeration of latent variable values
- Multiple levels of latent variables allow combinatorial sharing of statistical strength

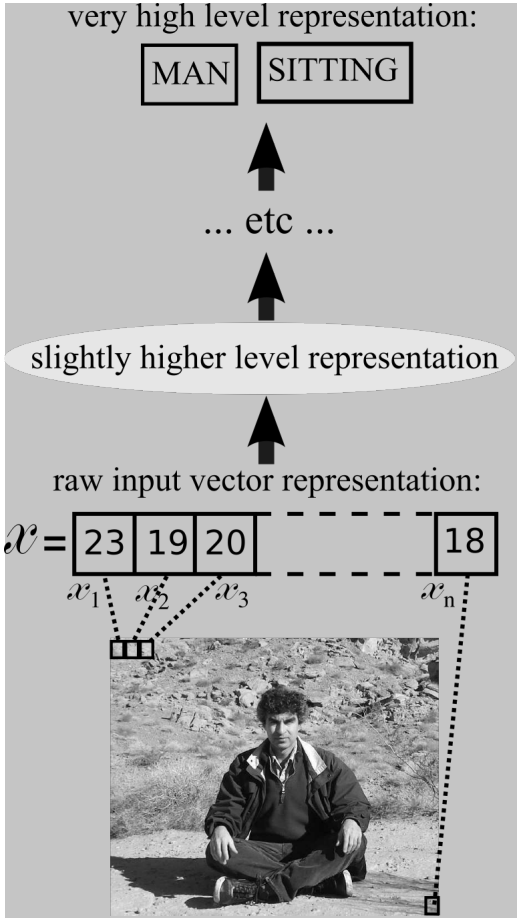




Deep Architecture in our Mind

- Humans organize their ideas and concepts hierarchically
- Humans first learn simpler concepts and then compose them to represent more abstract ones
- Engineers break-up solutions into multiple levels of abstraction and processing
- It would be nice to learn / discover these concepts

(knowledge engineering failed because of limits of introspection?)



Deep Learning Hypotheses

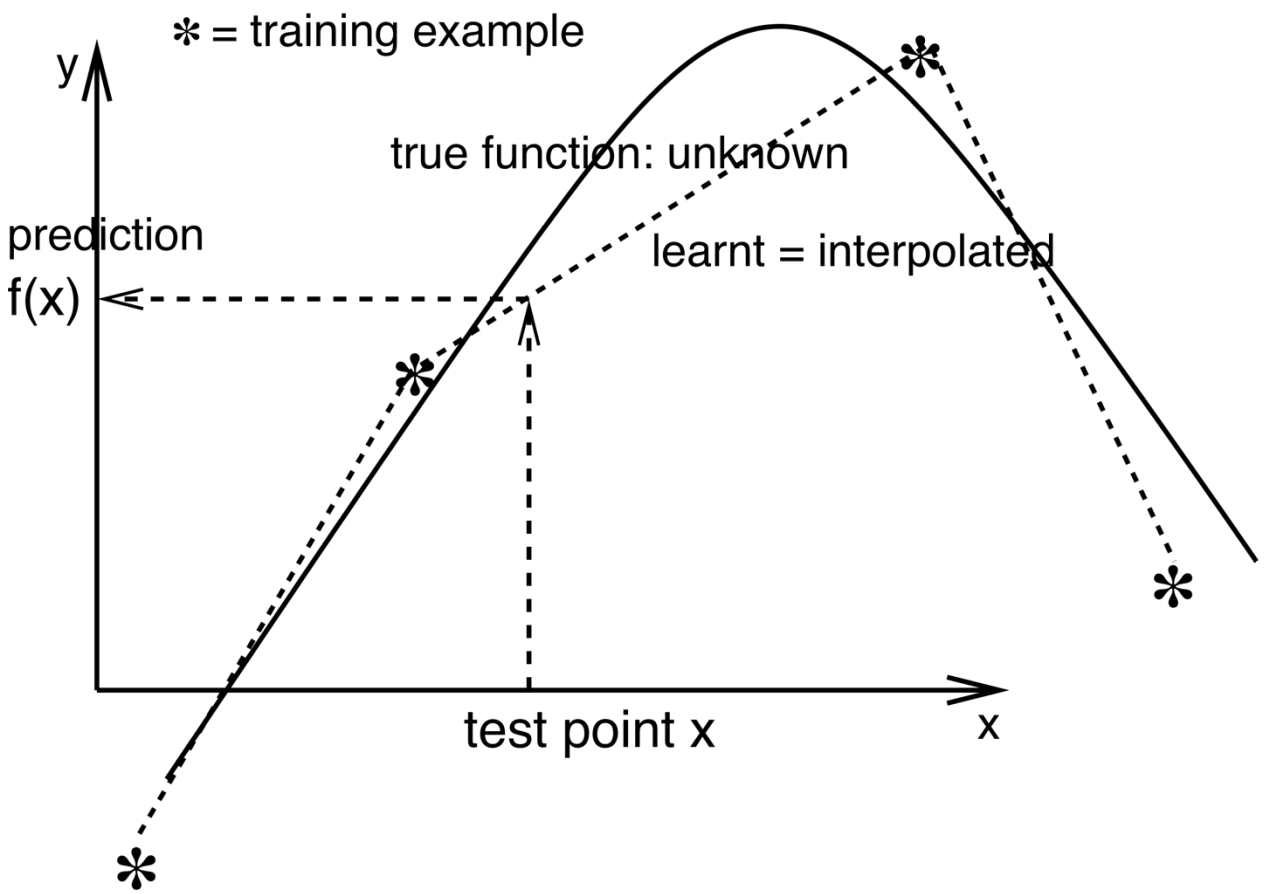
- Hypothesis 1: deep hierarchy of features useful to efficiently represent and learn complex abstractions needed for AI and mammal intelligence.
  - **Computational & statistical efficiency**

- Hypothesis 2: unsupervised **learning of representations** is a crucial component of the solution.
  - **Optimization & regularization.**

Even if we had labels for all examples

- Theoretical and ML-experimental support for both.

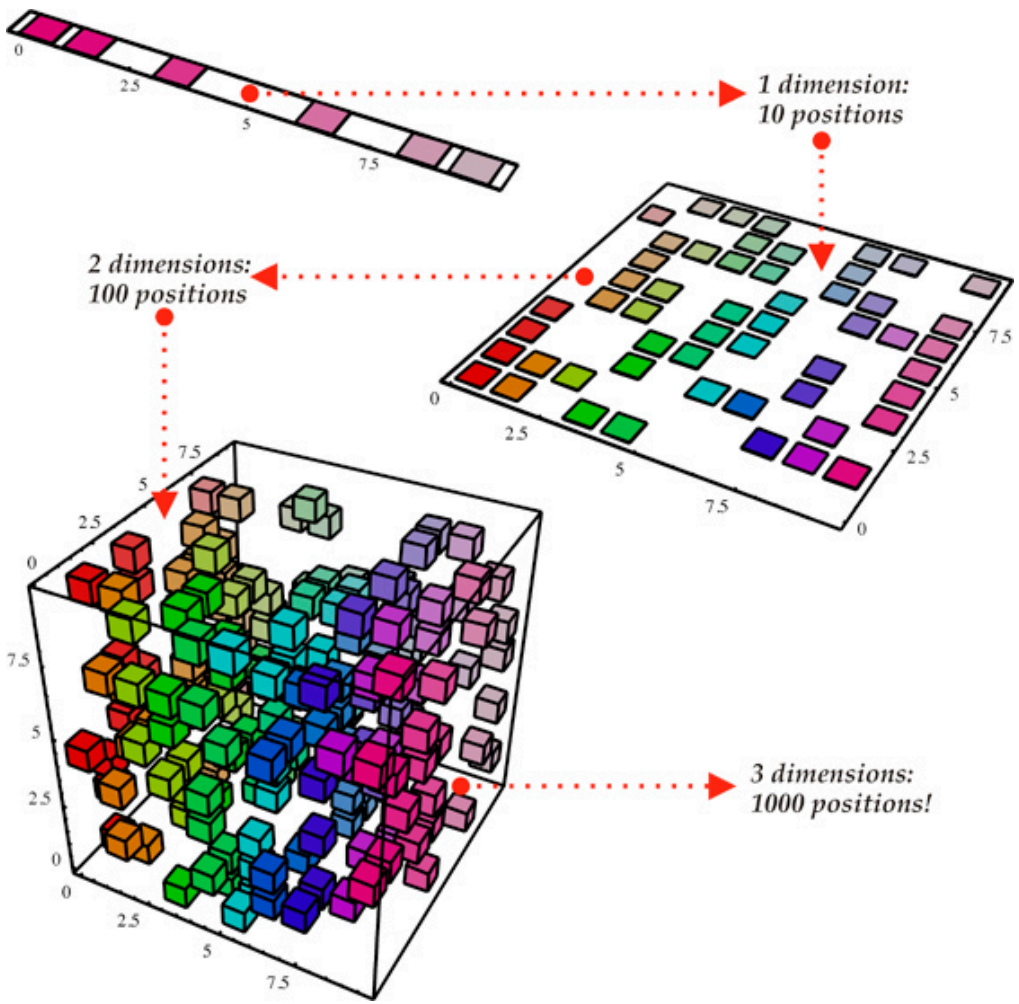
Principle of Local Generalization



The Curse of Dimensionality

To generalize locally, need representative examples for all relevant variations!

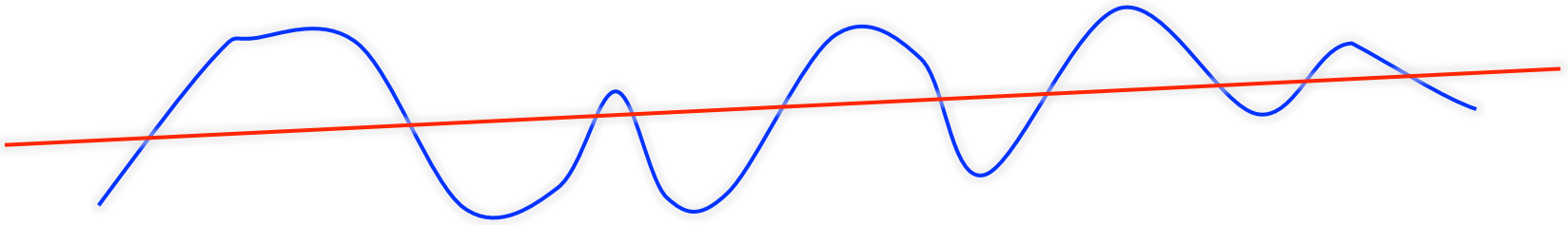
Classical solution: hope for a smooth enough target function



Limits of Local Generalization: Theoretical Results


e.g. Gaussian (RBF) SVM

- Theorem: Gaussian kernel machines need at least *k* examples to learn a function that has *2k* zero-crossings along some line



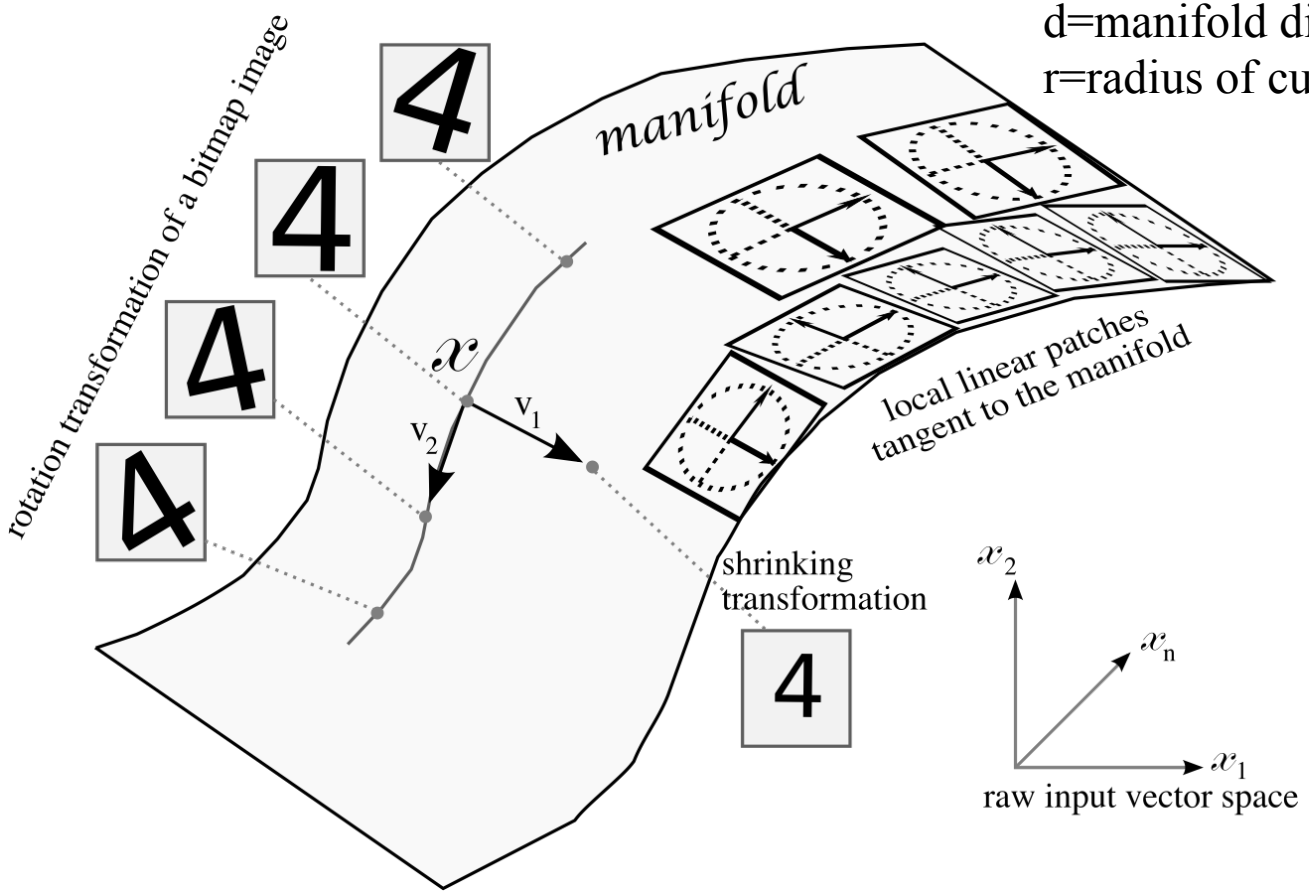
- Theorem: For a Gaussian kernel machine to learn some maximally varying functions over *d* inputs requires $O(2^d)$ examples

Curse of Dimensionality When Generalizing Locally on a Manifold

(Bengio et al 2006)

$O(d\, r^d)$ examples are needed

d=manifold dim.
r=radius of curvature



How to Beat the Curse of Many Factors of Variation?

Compositionality: exponential gain in representational power

- Distributed representations / embeddings: feature learning
- Deep architecture: multiple levels of feature learning

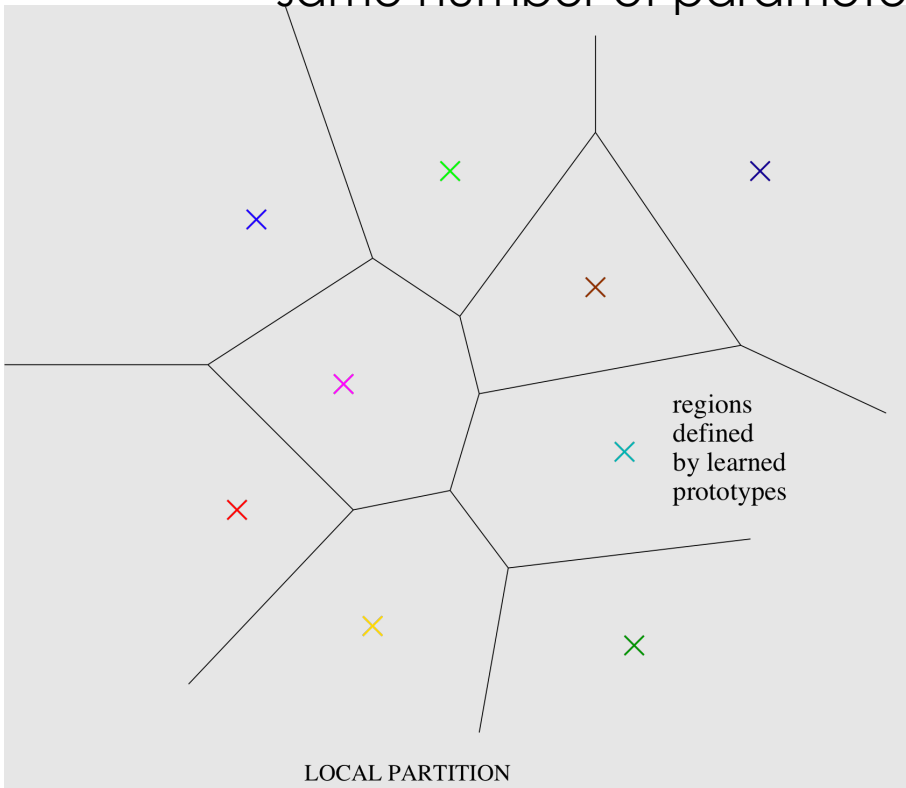
Can generalize to new configurations

Distributed Representations

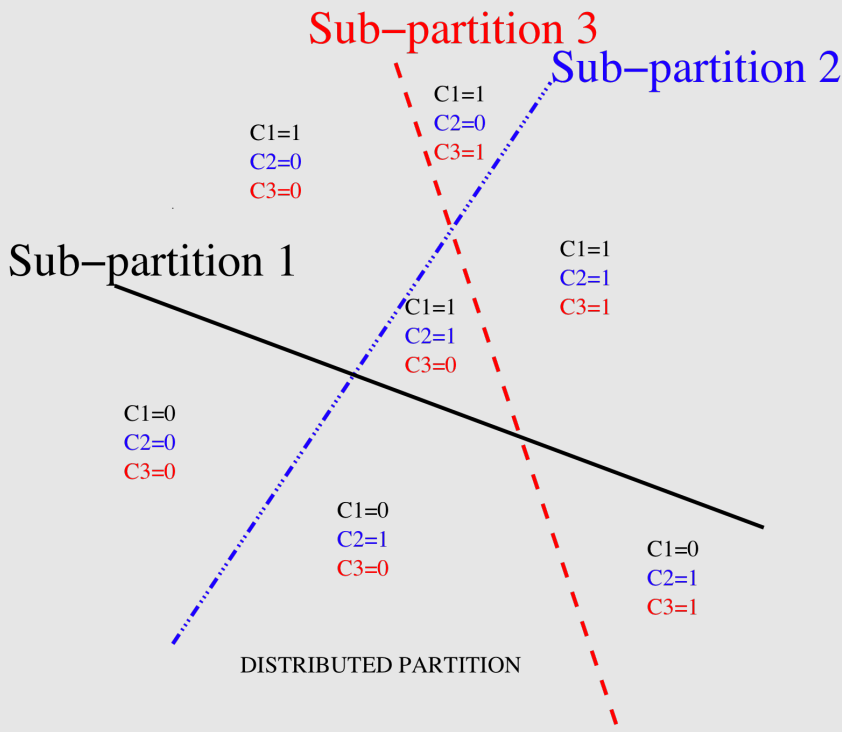
- Many features active simultaneously
- Input represented by the activation of a set of features that are not mutually exclusive
- Can be exponentially more efficient than local representations
- = FEATURE LEARNING instead of / on top of manual feature-engineering

Local vs Distributed Latent Variables

Exponentially more regions can be distinguished for the same number of parameters, i.e., examples

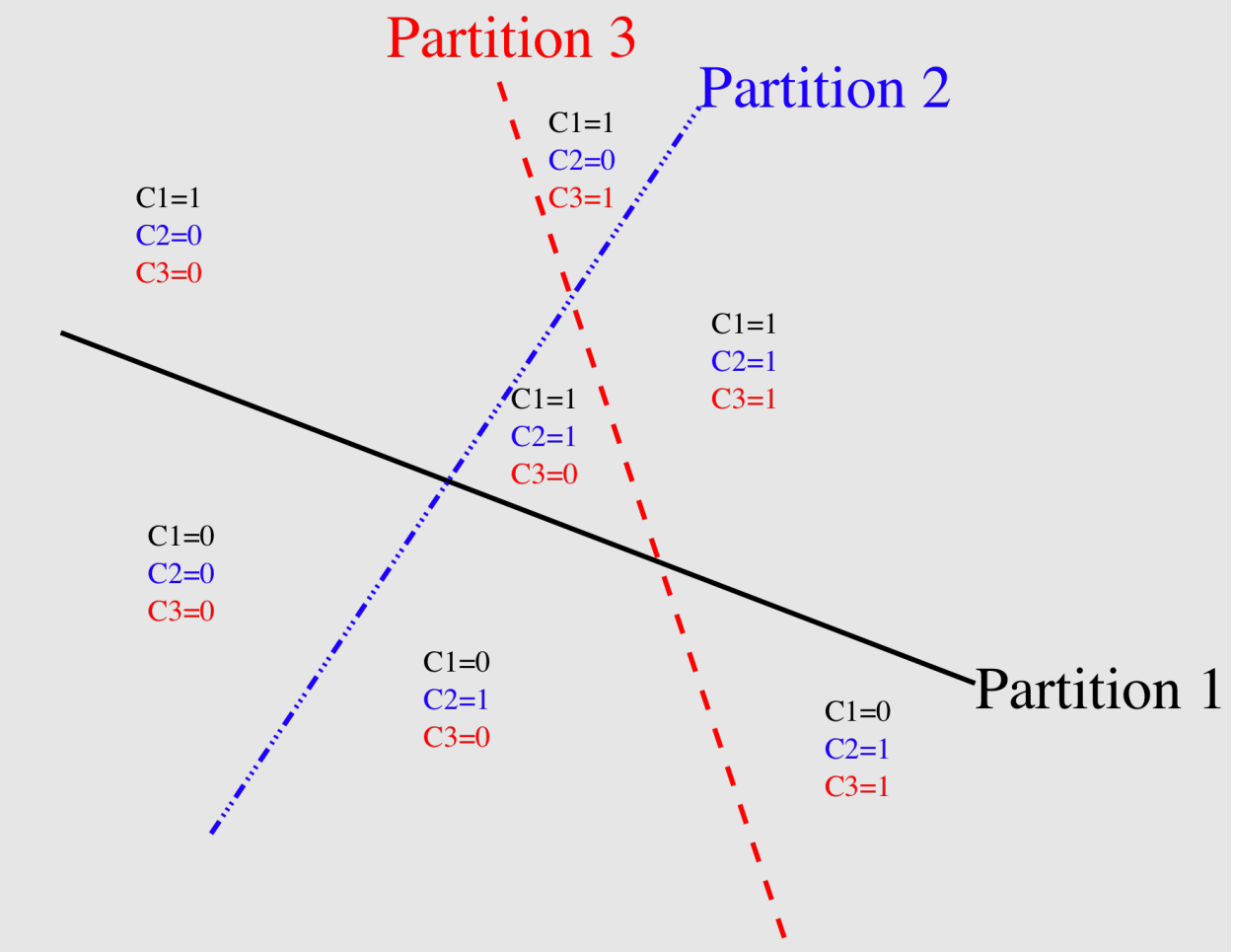


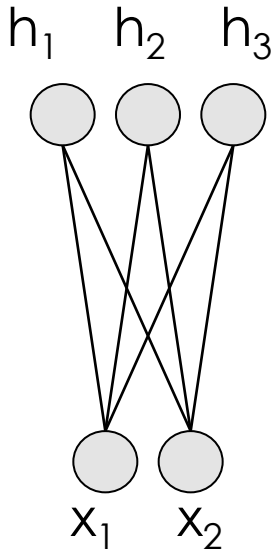
Clustering



Multi-clustering

RBM Hidden Units Carve Input Space

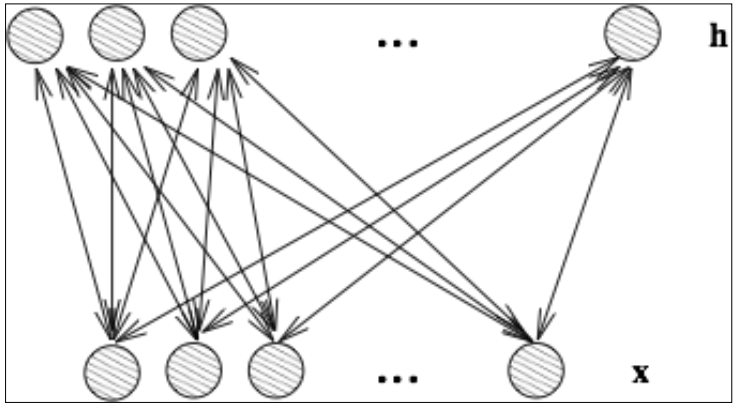




Restricted Boltzmann Machine

- The most popular building block for deep architectures

$$P(x,h) = \frac{1}{Z} e^{b^T h + c^T x + h^T W x}$$



Hidden units

Visible units

- Bipartite undirected graphical model
- Inference is trivial:
- P(h|x) & P(x|h) factorize

Discrete Input RBMs are Universal Approximators

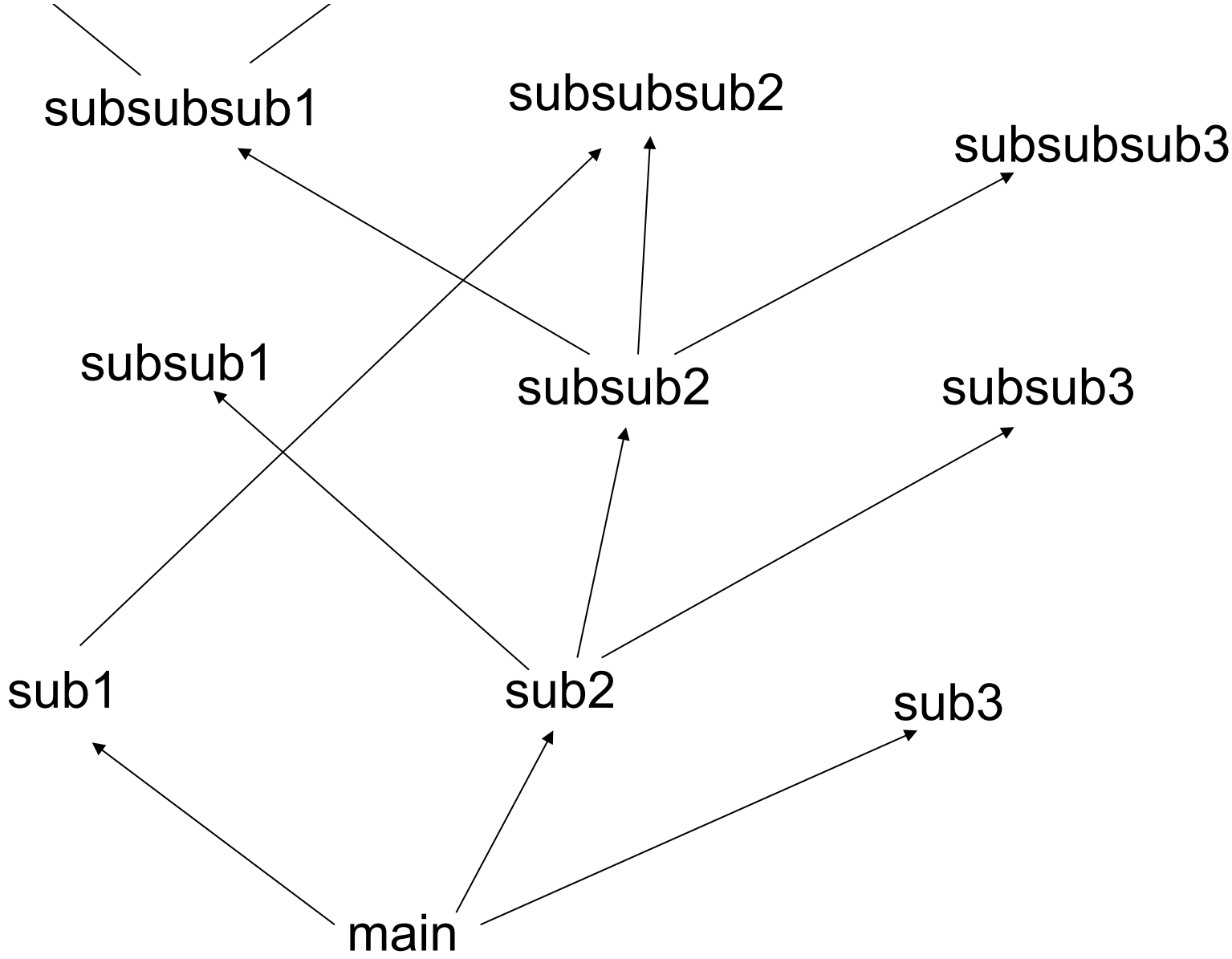
- Adding one hidden unit (with proper choice of parameters) guarantees increasing likelihood
- With enough hidden units, can perfectly model any discrete distribution
- RBMs with variable nb of hidden units = non-parametric

Continuous Inputs - The Best Generative Model of Images: Spike-and-Slab RBM

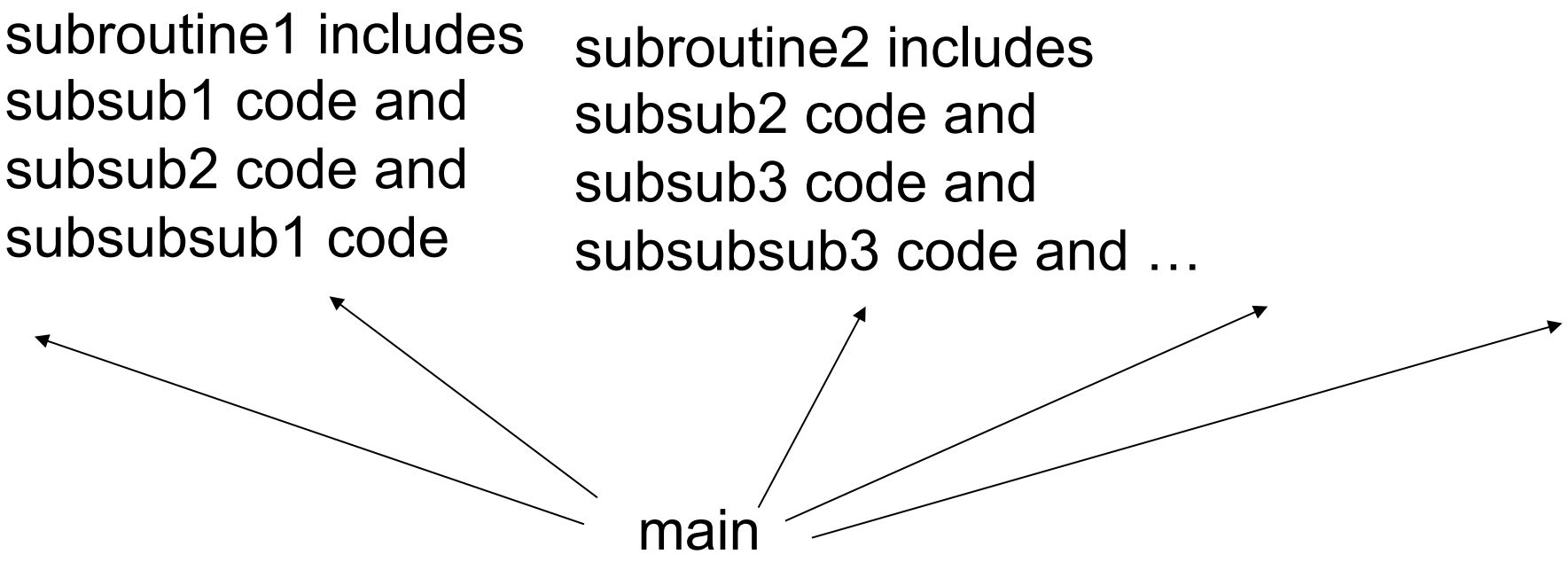
Samples from μ-ssRBM:

Nearest examples in CIFAR:
(least square dist.)

ICML 2011

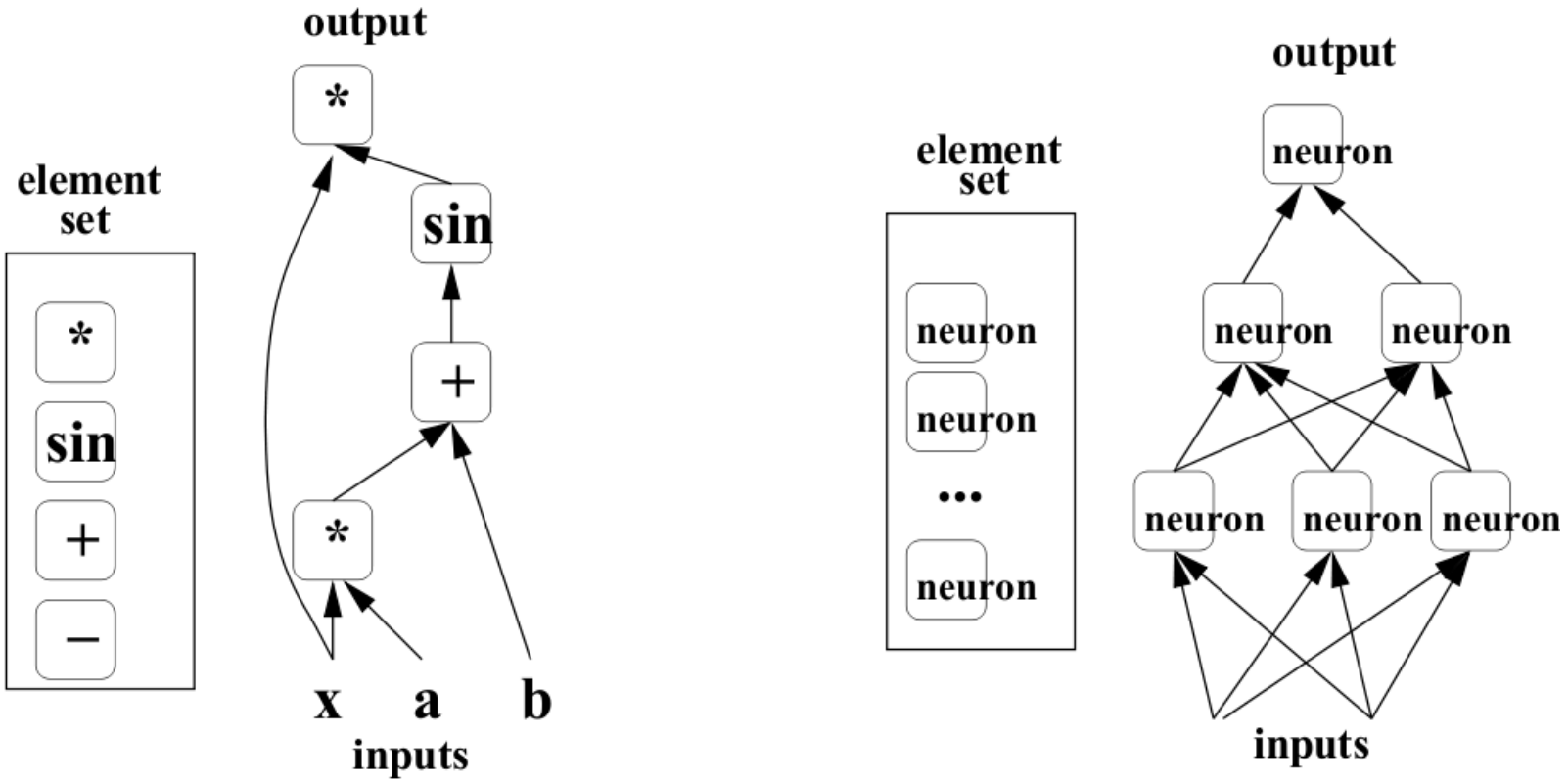


“Deep” computer program



“Shallow” computer program

Architecture Depth



“Deep” circuit

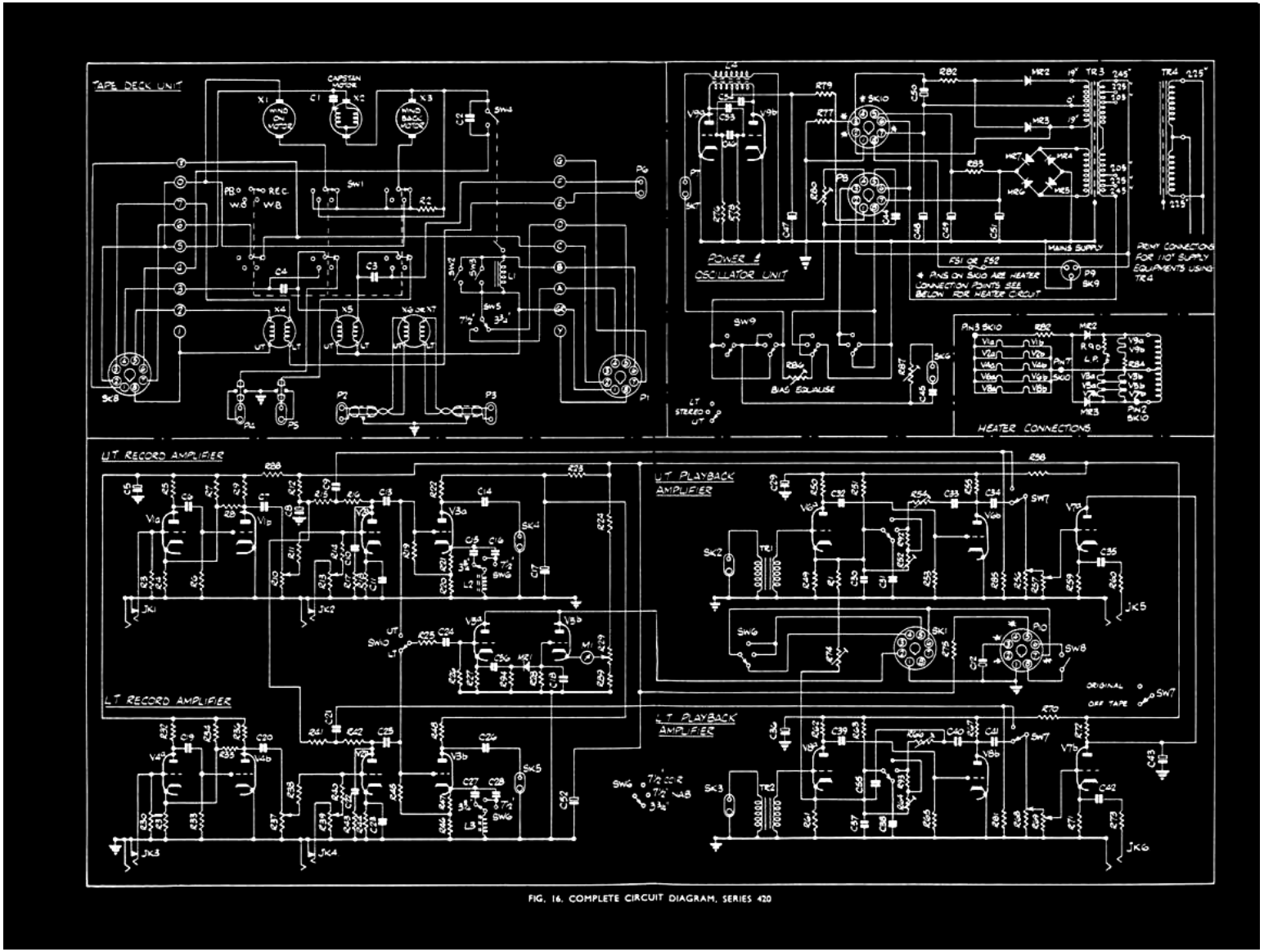
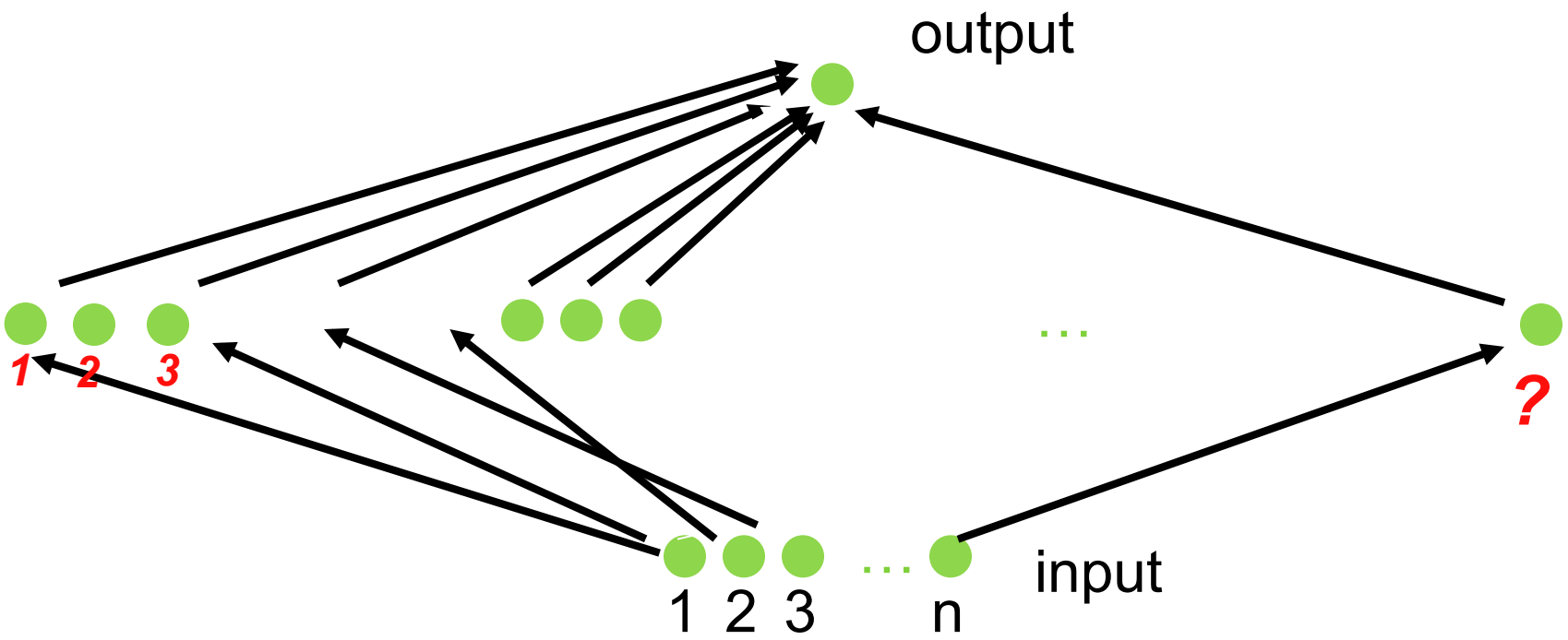


FIG. 16. COMPLETE CIRCUIT DIAGRAM, SERIES 420

“Shallow” circuit



Falsely reassuring theorems: one can approximate any reasonable (smooth, boolean, etc.) function with a 2-layer architecture

Deep Architectures are More Expressive

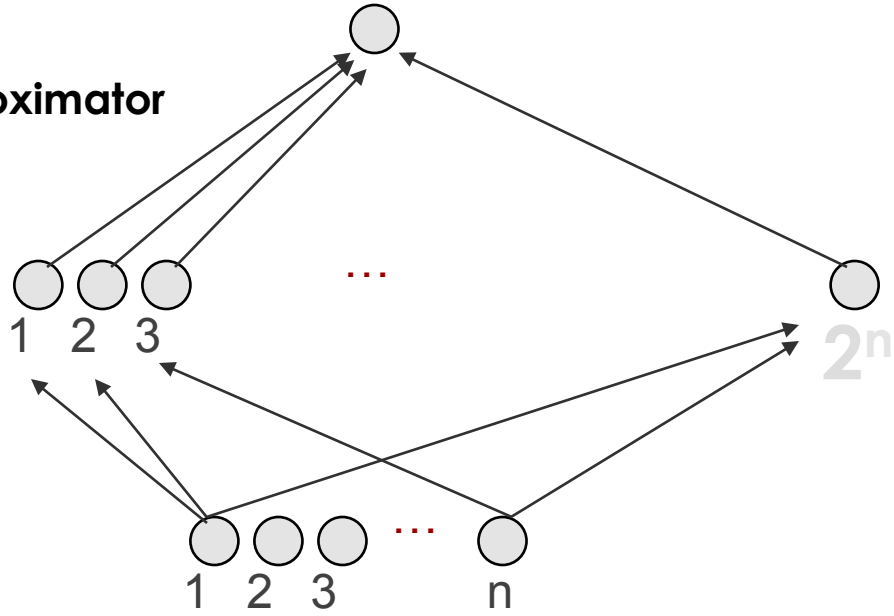

Theoretical arguments:

2 layers of { Logic gates / Formal neurons / RBF units } = universal approximator

RBMs & auto-encoders = universal approximator

Theorems on advantage of depth:
(**Hastad** et al 86 & 91, Bengio et al 2007, Bengio & Delalleau 2011, Braverman 2011)

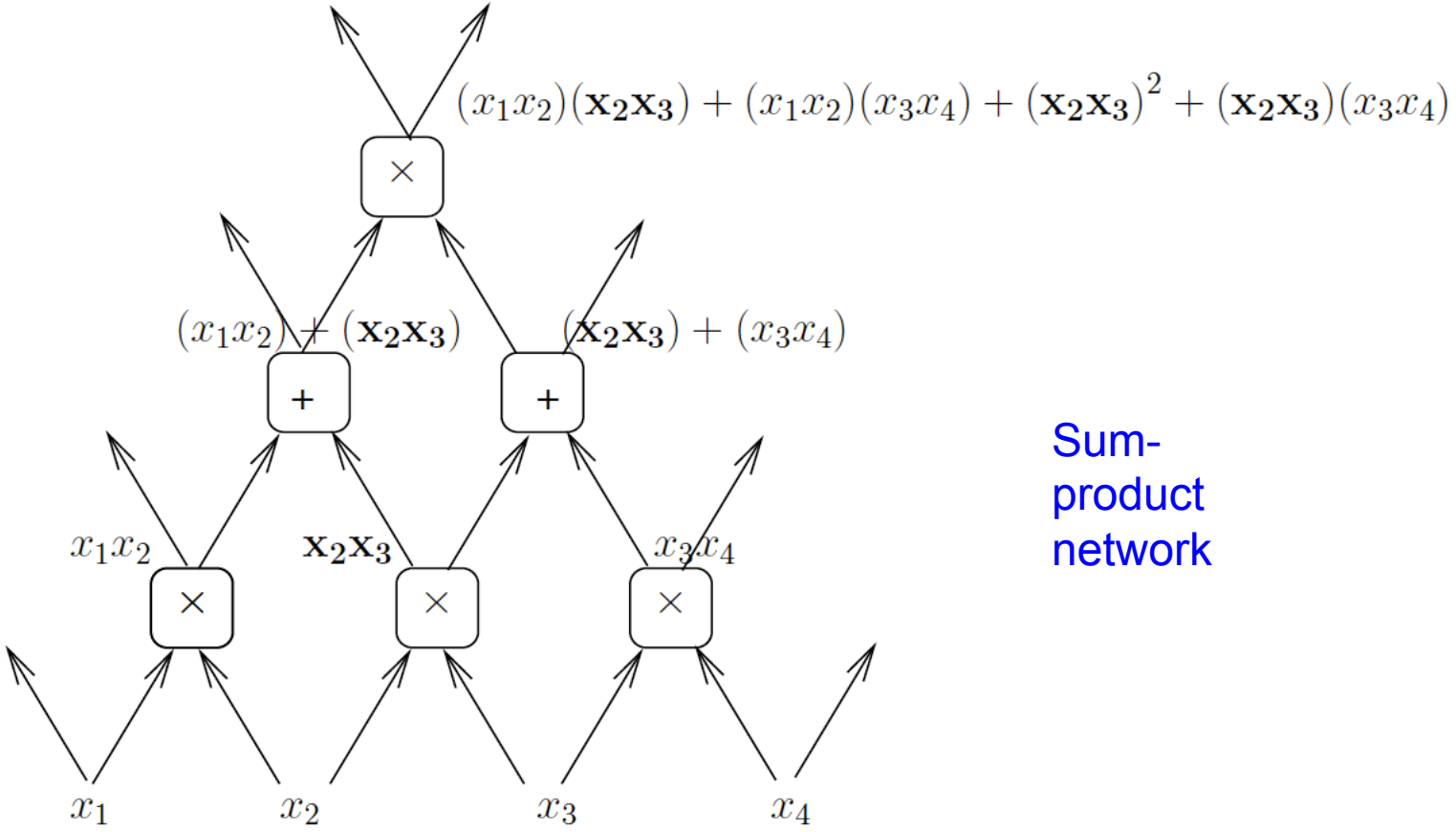
Functions compactly represented with k layers may require exponential size with insufficient depth



Sharing Components in a Deep Architecture

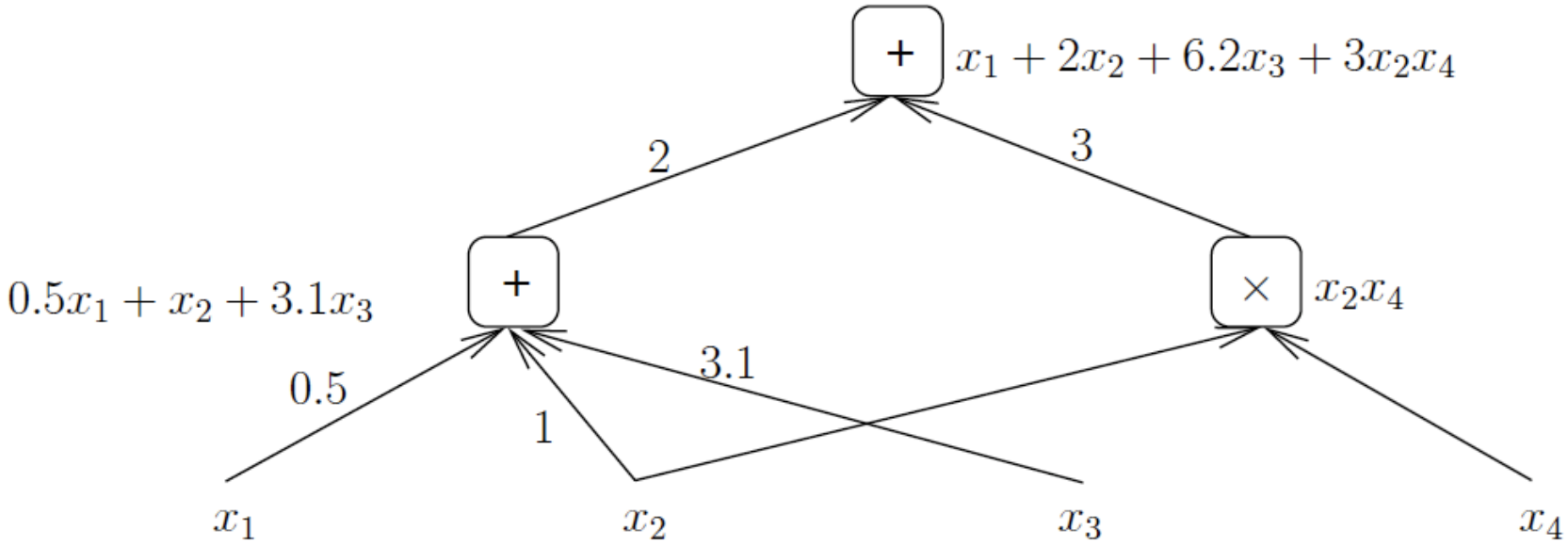
Polynomial expressed with shared components:

advantage of depth may grow exponentially



Sum-product network

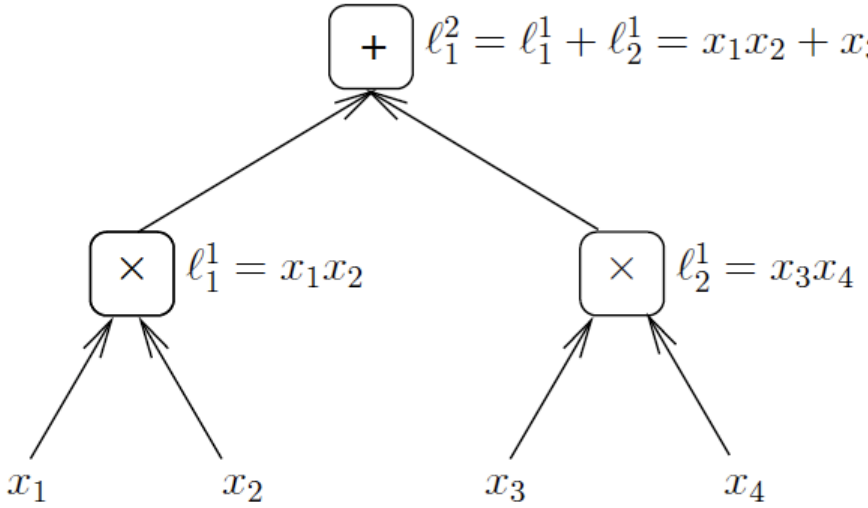
Sum-Product Networks



. Depth 2 suffices to represent any finite polynomial (sum of products)

. (Poon & Domingos 2010) use deep sum-product networks to efficiently parametrize partition functions

Polynomials that Need Depth

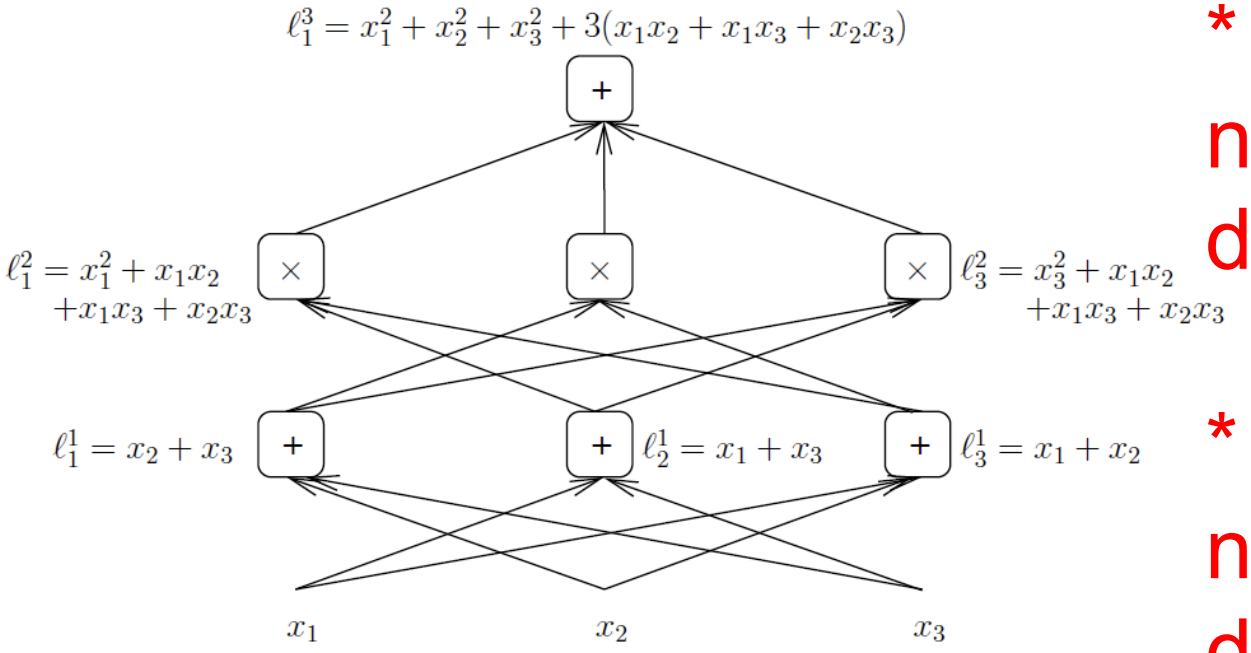


- $2i$ layers and $n = 4^i$ input variables
- alternate additive and multiplicative units
- unit ℓ_j^k takes as inputs ℓ_{2j-1}^{k-1} and ℓ_{2j}^{k-1}

* Need O(n) nodes with depth log(n) circuit

* Need $O(2^{\sqrt{n}})$ nodes with depth-2 circuit

More Polynomials that Need Depth



- $2i + 1$ layers and n variables (n independent of i)
- alternate multiplicative and additive units
- unit ℓ_j^k takes as inputs $\{\ell_m^{k-1} | m \neq j\}$

* Need O(dn) nodes with depth d circuit

* Need $O(n^d)$ nodes with depth-2 circuit

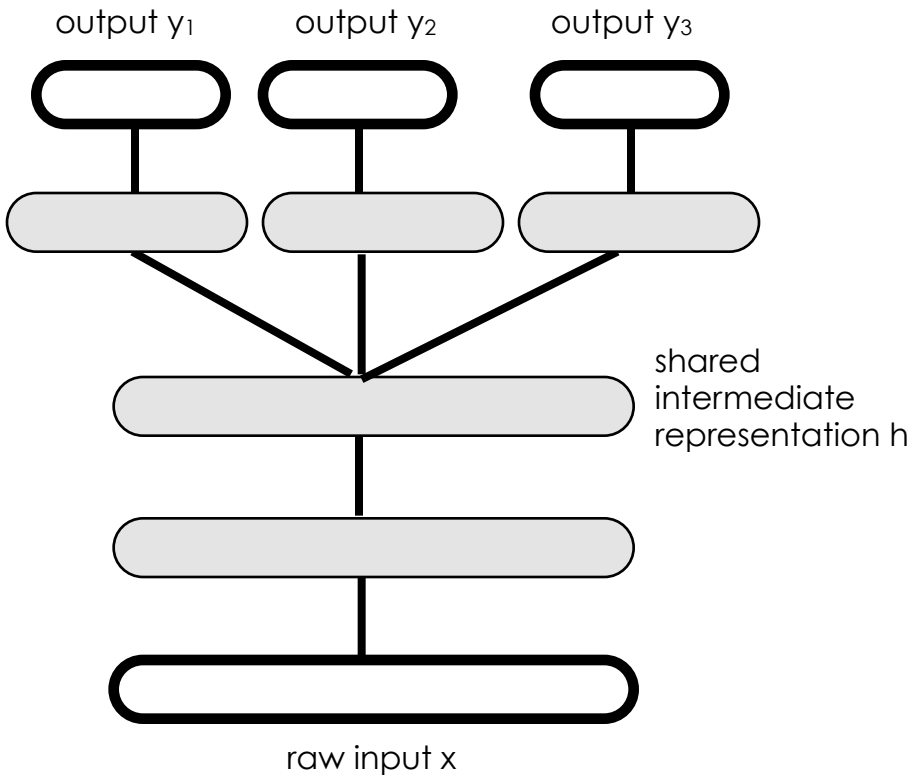
More Deep Theory

Poly-logarithmic Independence Fools Bounded-Depth Boolean Circuits
Braverman, CACM 54(4), April 2011.

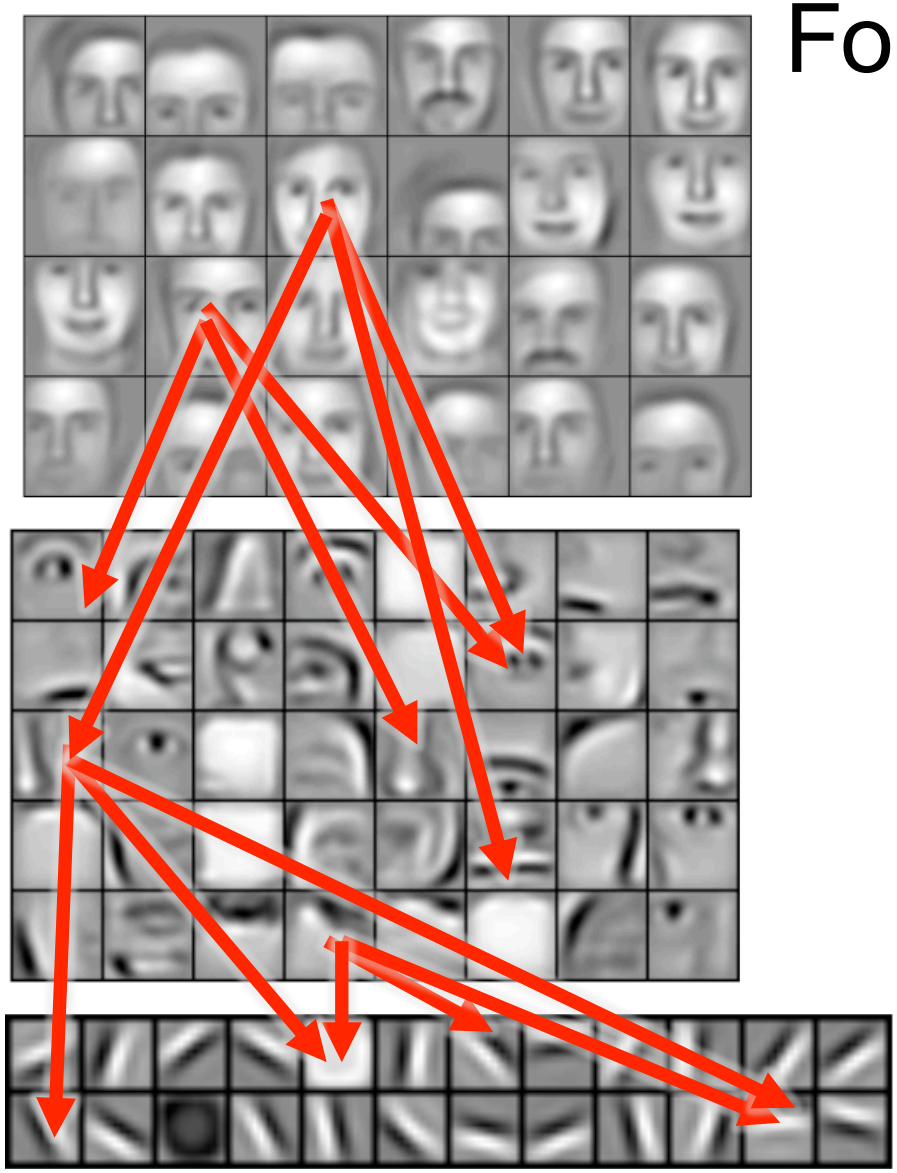
If all marginals of the input distribution involving at most k variables are uniform, higher depth makes it exponentially easier to distinguish the joint from the uniform.

Deep Architectures and Sharing Statistical Strength, Multi-Task Learning

- Generalizing better to new tasks is crucial to approach AI
- Deep architectures learn good intermediate representations that can be shared across tasks
- Good representations make sense for many tasks



Parts Are Re-Used to Form Different Objects



(Lee et al. ICML 2009)

Layer 3: objects

Layer 2: parts

Layer 1: edges

Before 2006

Failing to train deep architectures

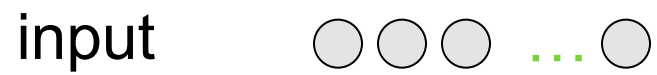
2006

Breakthrough!

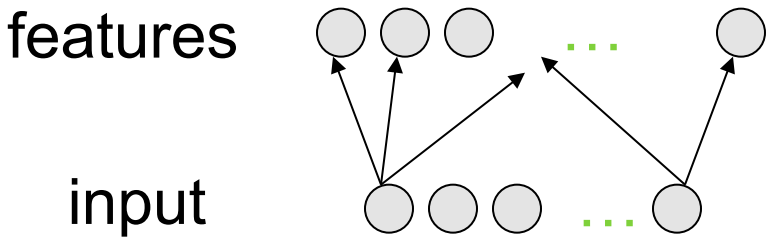
2006: The Deep Breakthrough

- Hinton, Osindero & Teh « A Fast Learning Algorithm for Deep Belief Nets », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », *NIPS'2006*

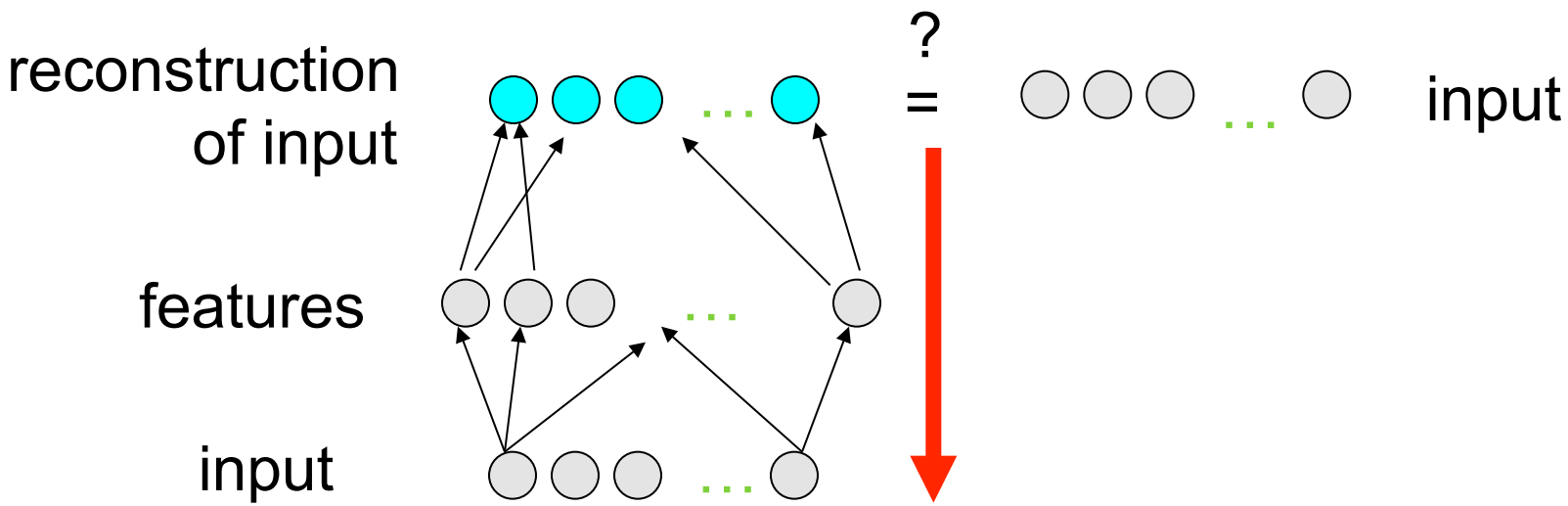
Deep training



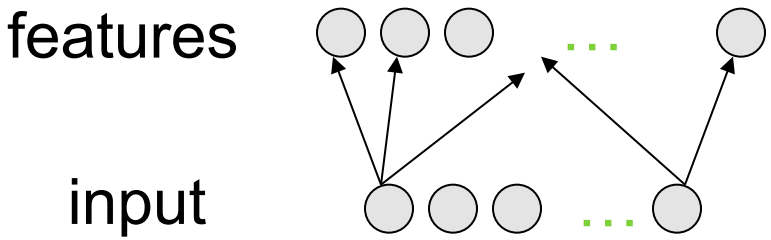
Layer-Wise Unsupervised Pre-training



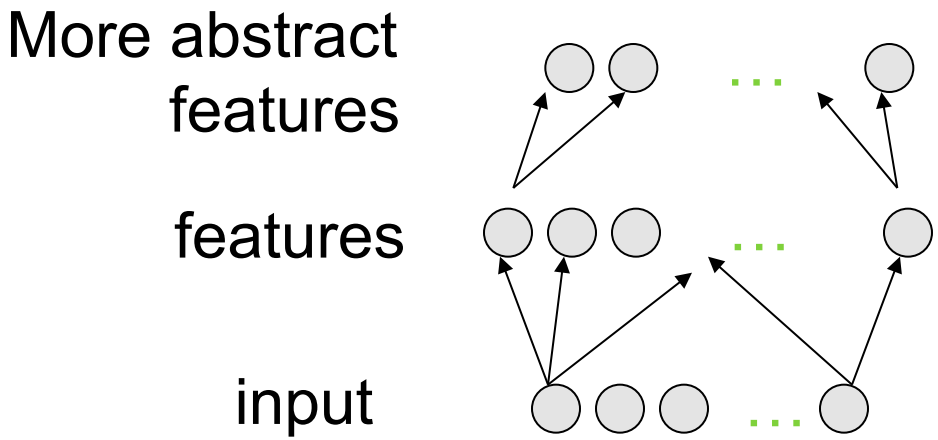
Layer-Wise Unsupervised Pre-training



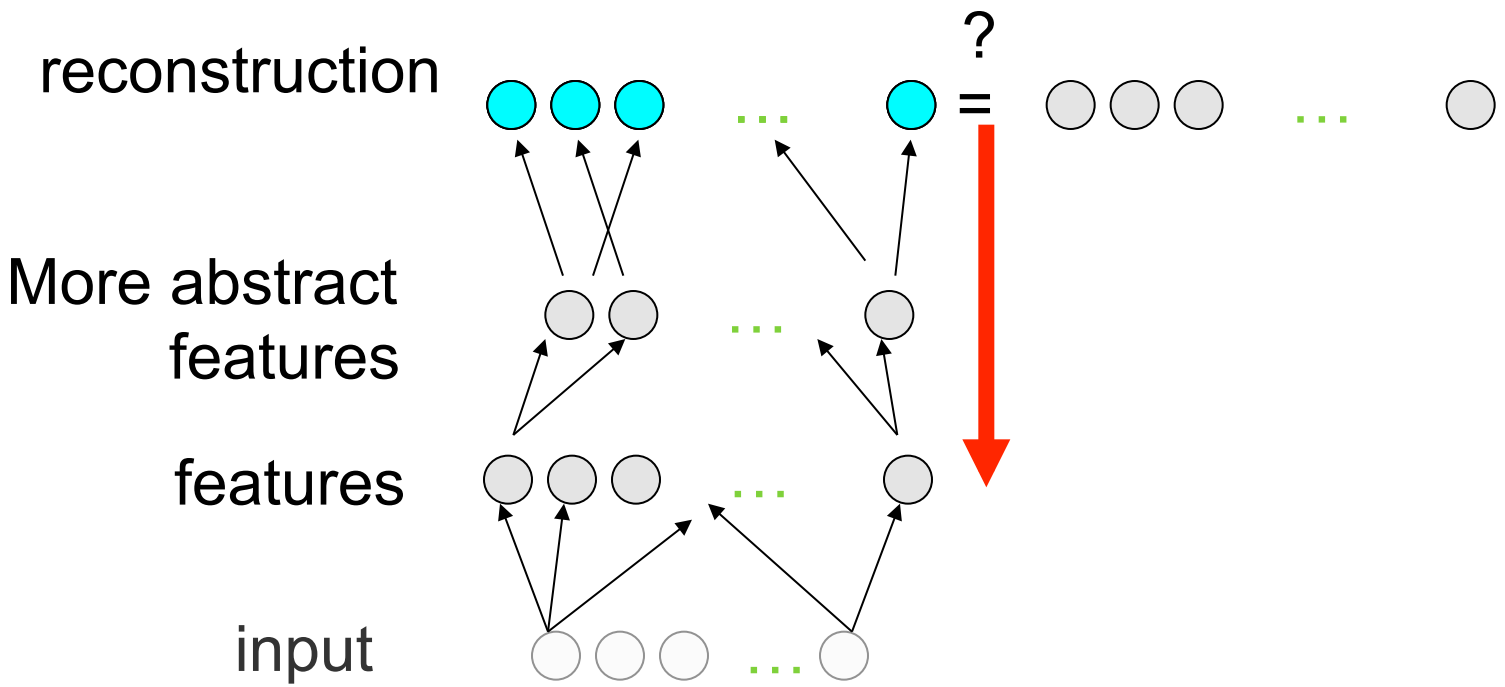
Layer-Wise Unsupervised Pre-training



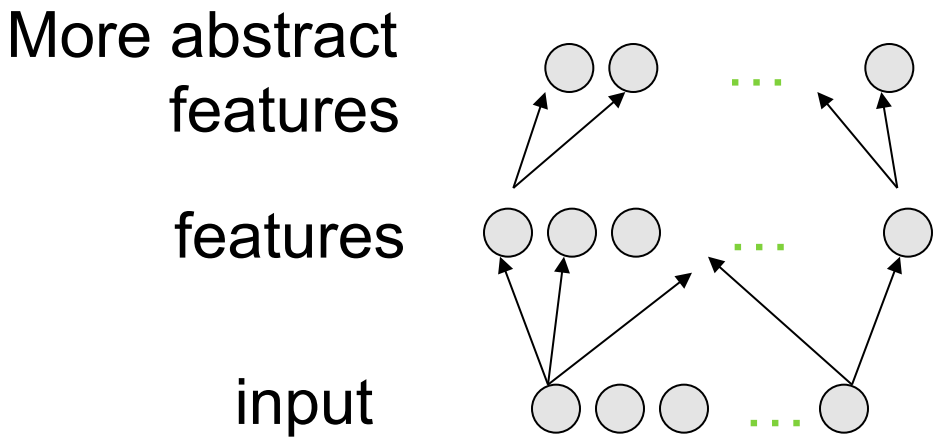
Layer-Wise Unsupervised Pre-training



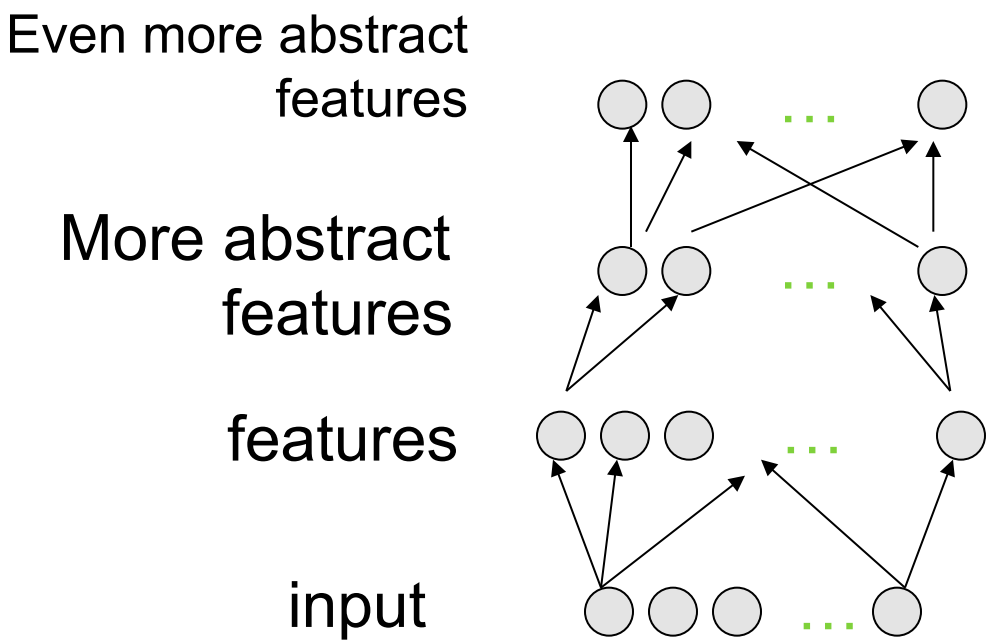
Layer-Wise Unsupervised Pre-training



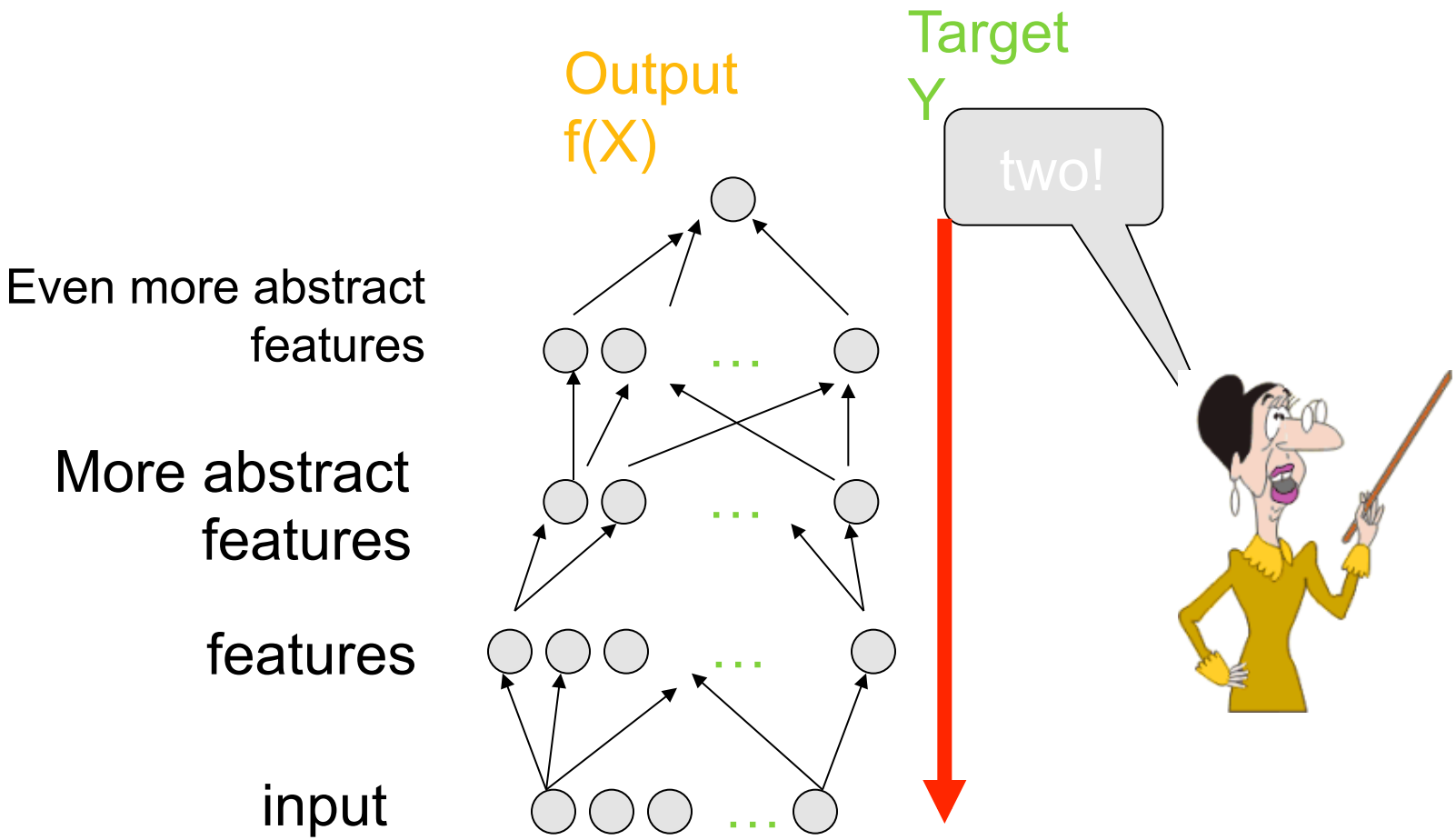
Layer-Wise Unsupervised Pre-training



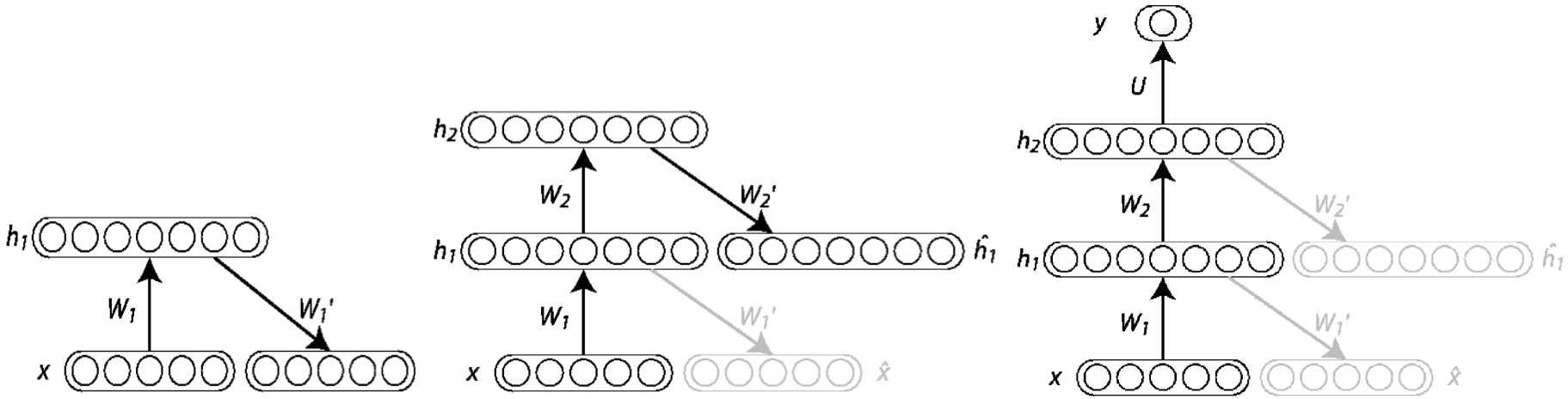
Layer-Wise Unsupervised Pre-training



Supervised Fine-Tuning



Stacking Auto-Encoders



Palette of Tricks to Train Energy-Based Models

Partition function expensive (vocab.) or intractable

- Contrastive Divergence
- PCD / SML + MCMC tricks
  - Tempering
  - Mean-field / variational, etc.
- (regularized) Score Matching / denoising
- Sparse coding / Sparse Predictive Decomposition
- Ratio Matching
- Pseudo-likelihood
- Ranking / margin-based criteria
- Noise contrastive estimation

- **Most rely on + vs – examples contrast**

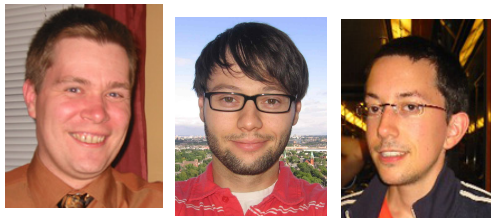
See my book / review paper (F&TML 2009): Learning Deep Architectures for AI

Sparse Auto-Encoders & Sparse Coding

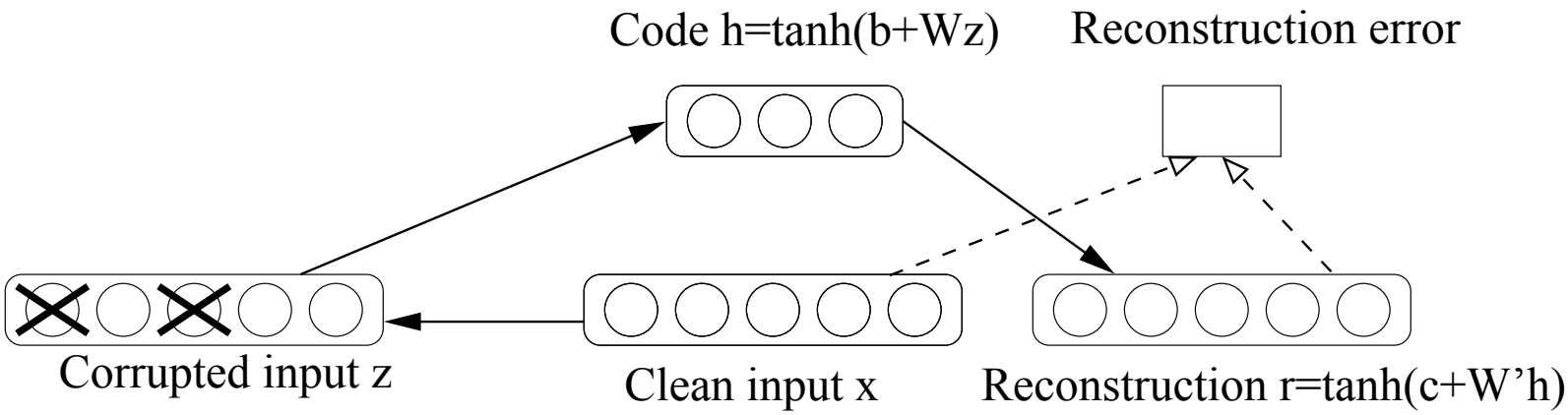
- Penalty on the representation to achieve sparsity.
- Stacked sparse auto-encoders successfully used by Andrew Ng's group at Stanford (e.g. ICML 2011)
- Used by Google in their Google Goggles vision system
- Sparse coding (recently stacked as well)
- Sparse Predictive Decomposition (LeCun)

Denoising Auto-Encoder

(Vincent et al 2008, 2010)

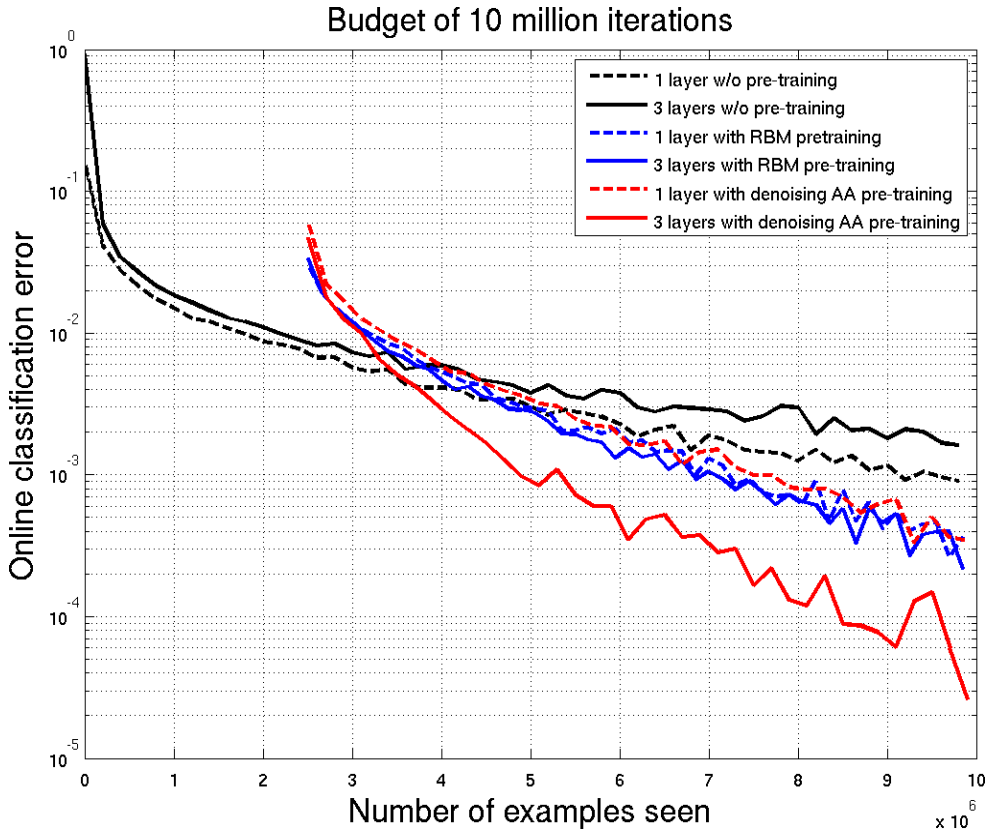


- Stochastically corrupt the input
- Reconstruction target = clean input

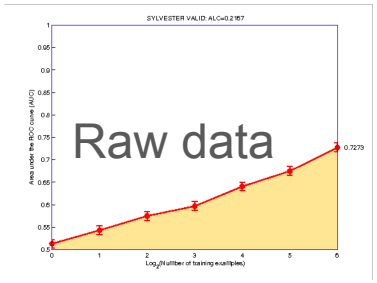


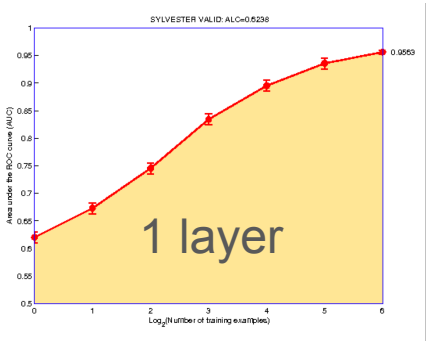
Stacked Denoising Auto-Encoders

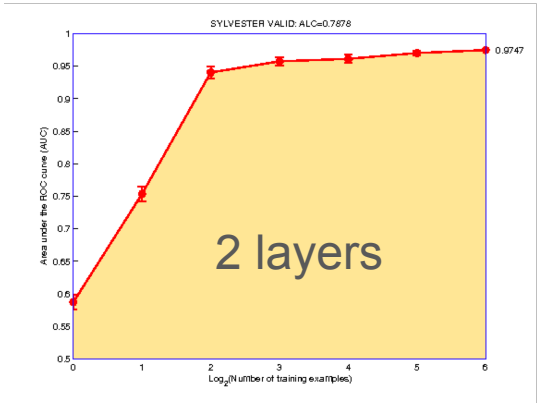
- No partition function, can measure training criterion
- Encoder & decoder: any parametrization
- As good or better than RBMs for feature learning
- = regularized score matching



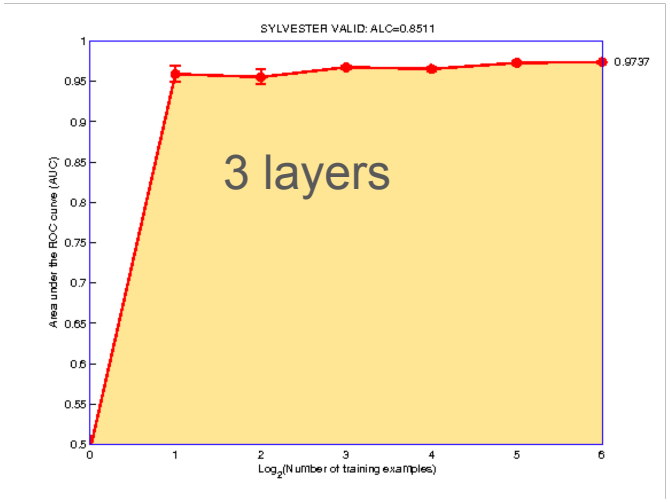
Unsupervised and Transfer Learning Challenge: 1st Place in Final Phase

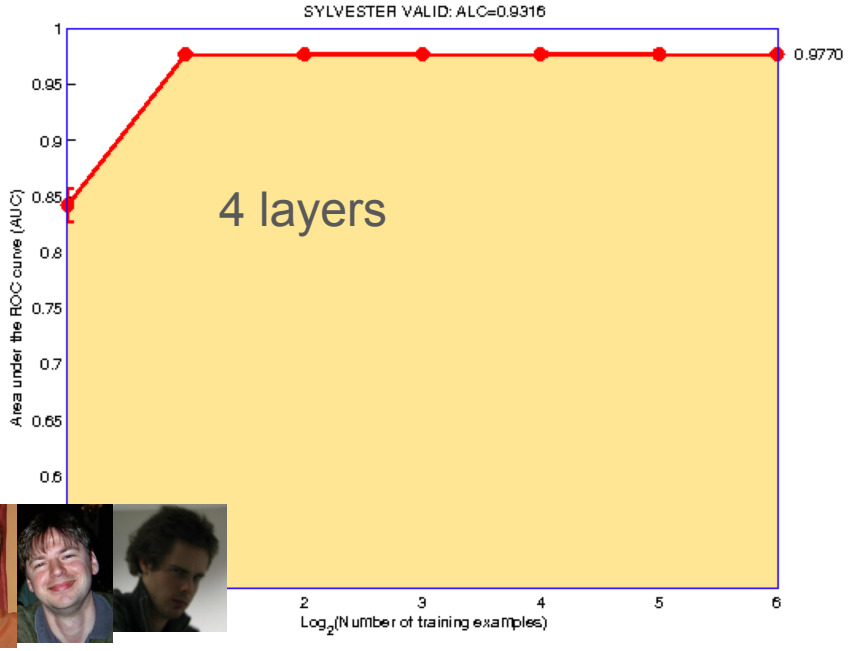




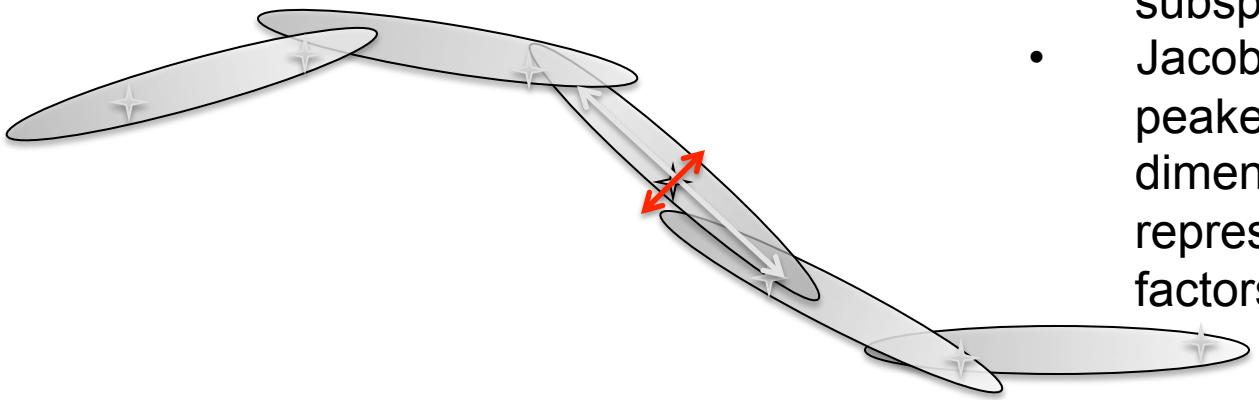


ICML'2011 workshop on UTL





Contractive Auto-Encoders



- Few active units represent the active subspace local chart
- Jacobian's spectrum is peaked = local low-dimensional representation / relevant factors

cannot afford contraction in manifold directions

wants contraction in all directions

Training criterion:

$$\mathcal{J}_{\mathrm{CAE}}(\theta) = \sum_{x \in D_n} \left(L(x, g(h(x))) + \lambda \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i} \right)^2 \right.$$

Manifold Tangent Classifier (NIPS 2011)

- Leading singular vectors on MNIST, CIFAR-10, RCV1:

Trading & Markets	+gilt	-slow	+matur	-percent	+bln	-anti	+interest	-sen
	+yen	-term	+auction	-sent	+coupon	-predict	+calcul	-californ
	+usda	-debt	+treasur	-pressure	+discount	-belgian	+overnight	-introduc

- **Knowledge-free MNIST: 0.81% error**

K-NN	NN	SVM	DBN	CAE	DBM	CNN	MTC
3.09%	1.60%	1.40%	1.17%	1.04%	0.95%	0.95%	**0.81**%

Unsupervised Learning: Disentangling Factors of Variation

- *(Goodfellow et al NIPS'2009)*: some hidden units more invariant (with more depth) to input geometry variations
- *(Glorot et al ICML'2011)*: some hidden units specialize on one aspect (domain) while others on another (sentiment)
- We don't want invariant representations because it is not clear to what aspects, but disentangling factors would help a lot
- Sparse/saturated units seem to help
- Why?
- **How to train more towards that objective?**

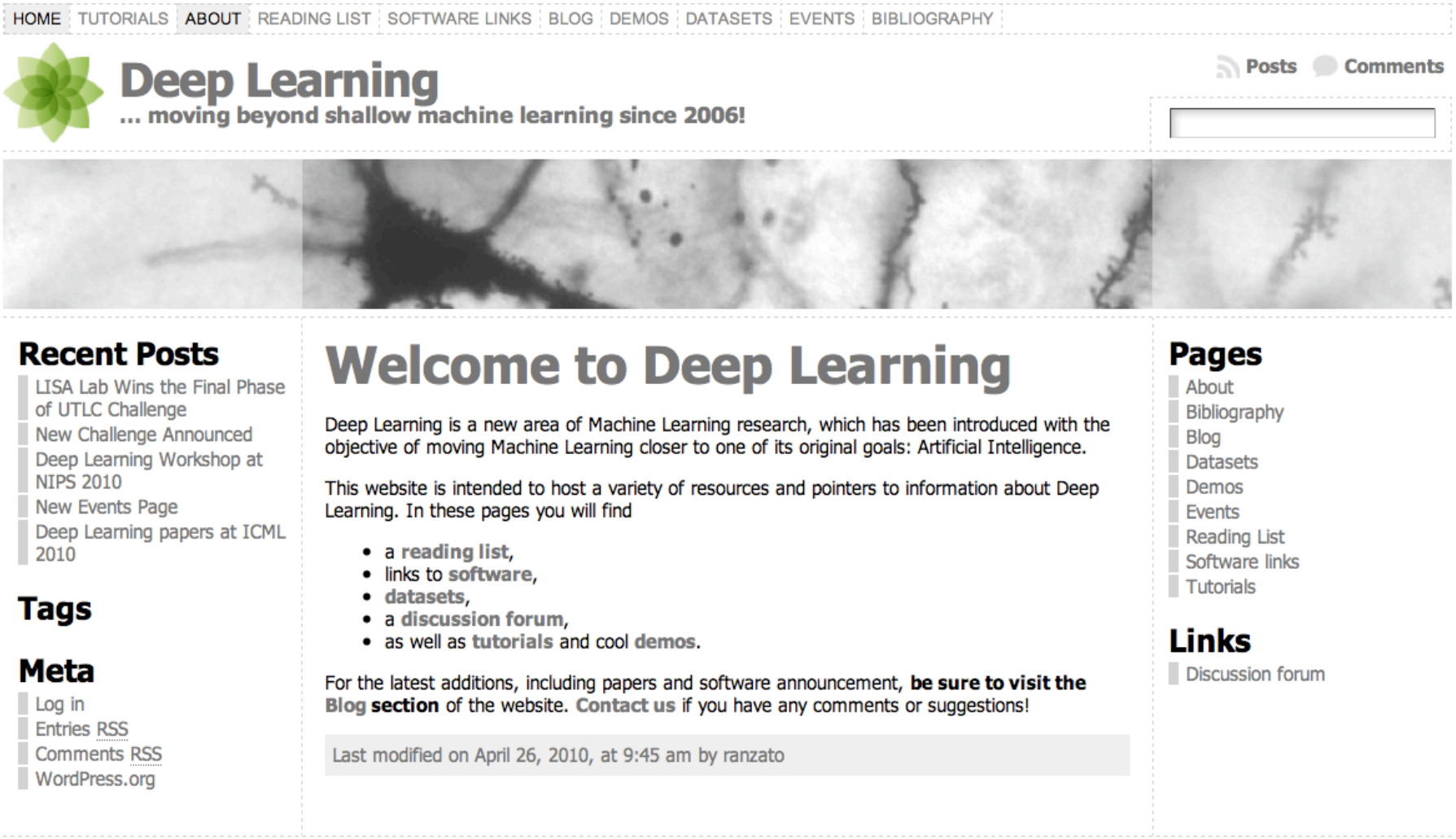
Recent Deep Learning Highlights

- Google Goggles uses stacked sparse auto-encoders (Hartmut Neven @ ICML 2011)
- UofT breaks old accuracy ceiling in TIMIT phoneme detection
- Stanford breaks records in video / gesture classification
- NYU breaks records in traffic sign class
- Montreal wins Unsupervised & Transfer Learning Challenge

Conclusions

- Deep Learning: powerful arguments & generalization principles
- Unsupervised Feature Learning is crucial: many new algorithms and applications in recent years
- DL particularly suited for multi-task learning, transfer learning, domain adaptation, self-taught learning, and semi-supervised learning with few labels

http://deeplearning.net



http://deeplearning.net/software/theano : numpy → GPU