Representation Learning and Deep Learning

Yoshua Bengio

Institute for Pure and Applied Mathematics (IPAM)
Graduate Summer School 2012 on deep learning
and feature learning

July 2012, UCLA

Université de Montréal

LISA

Outline of the Tutorial

1. Motivations and Scope
2. Algorithms
3. Analysis, Issues and Practice
4. Applications to NLP
5. Culture vs Local Minima

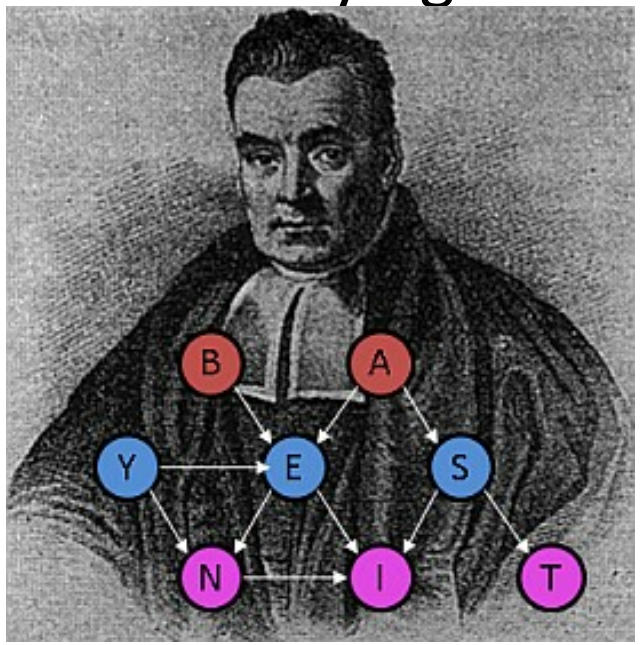
See (Bengio, Courville & Vincent 2012)
"Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives"
And http://www.iro.umontreal.ca/~bengioy/talks/deep-learning-gss2012.html for a pdf of the slides and a detailed list of references.

Ultimate Goals

- AI
- Needs knowledge
- Needs learning
- Needs generalizing where probability mass concentrates
- Needs ways to fight the curse of dimensionality
- Needs disentangling the underlying explanatory factors ("making sense of the data")

Representation Learning

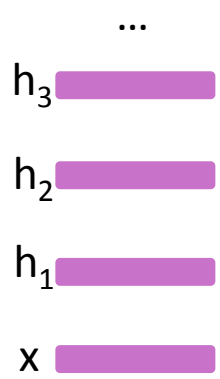
- Good features essential for successful ML
- Handcrafting features vs learning them
- Good representation: captures posterior belief about explanatory causes, disentangles these underlying factors of variation
- Representation learning: guesses the features / factors / causes = good representation.



Deep Representation Learning

Deep learning algorithms attempt to learn multiple levels of representation of increasing complexity/abstraction

When the number of levels can be data-selected, this is a deep architecture



A Good Old Deep Architecture

Optional Output layer

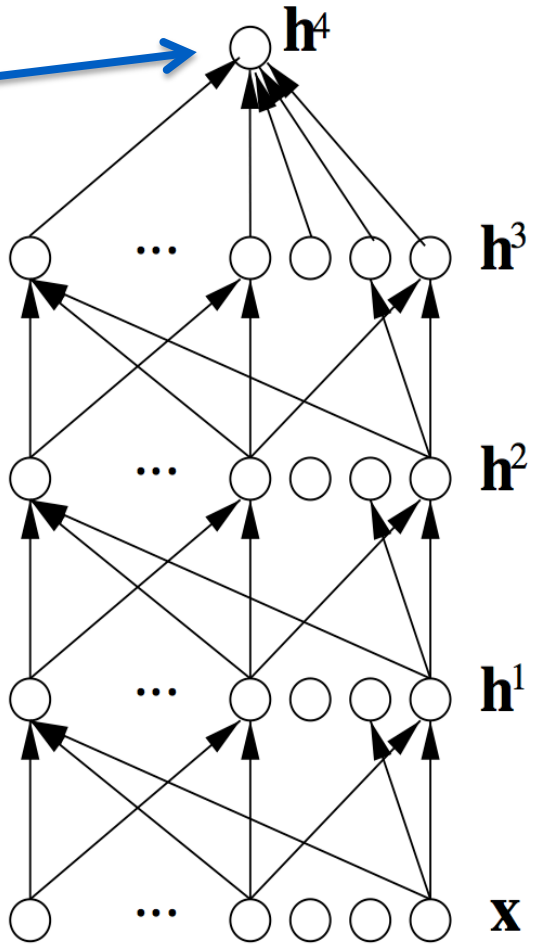
Here predicting a supervised target

Hidden layers

These learn more abstract representations as you head up

Input layer

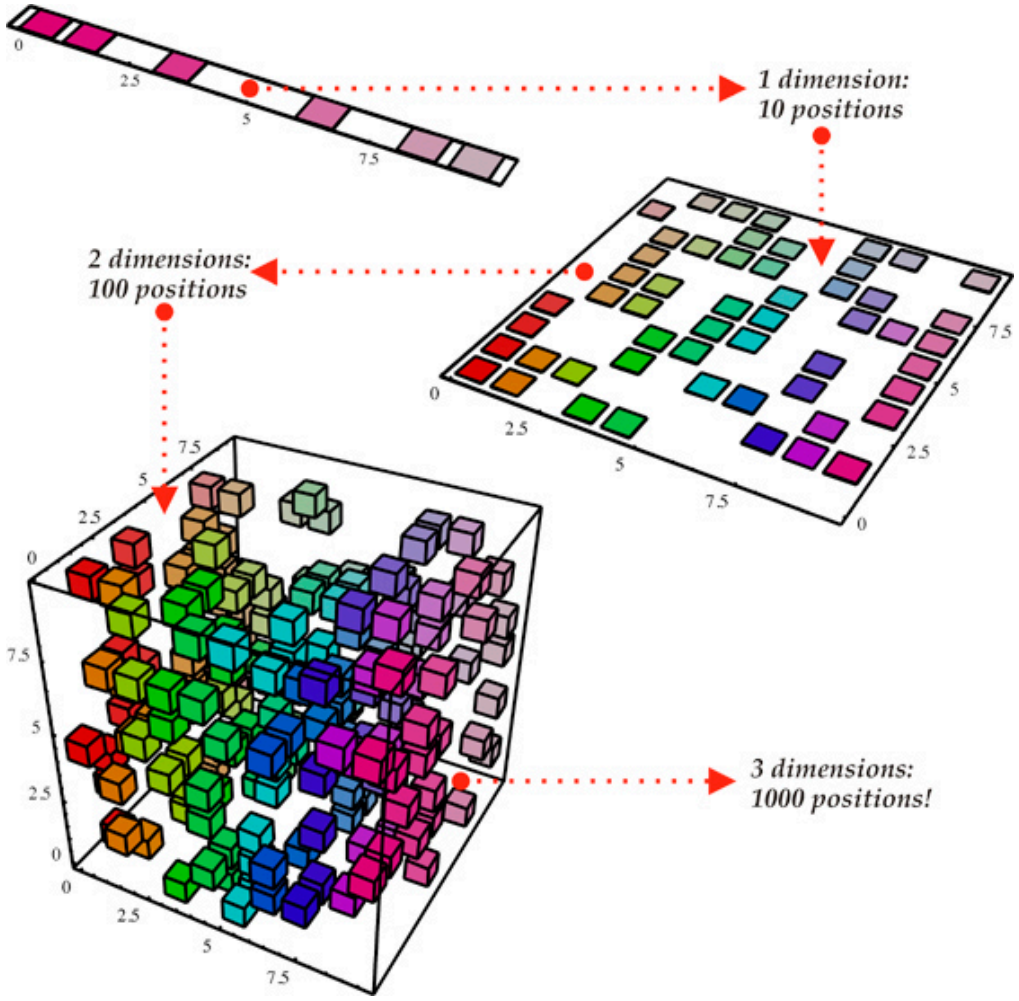
This has raw sensory inputs (roughly)



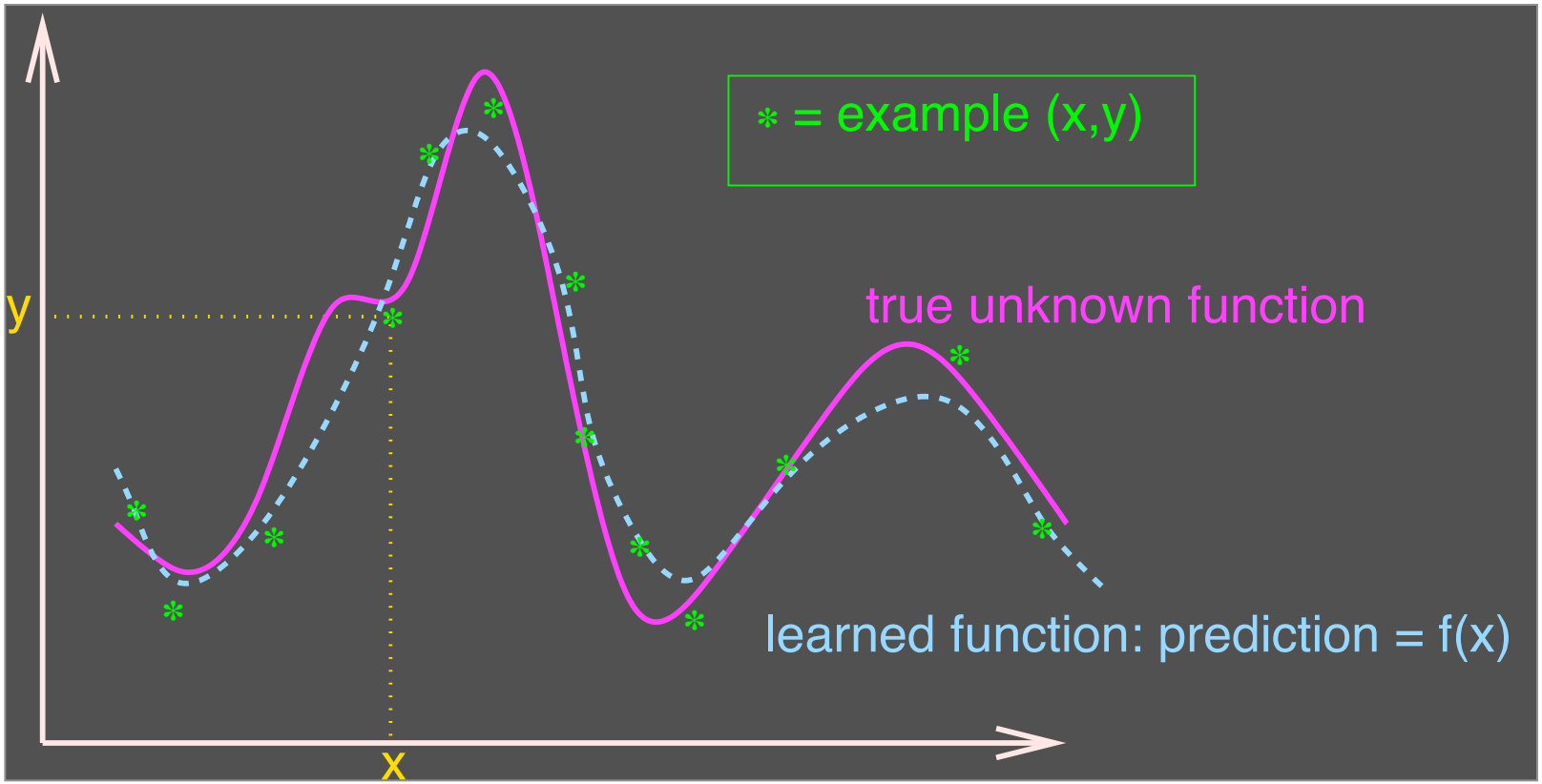
What We Are Fighting Against: The Curse of Dimensionality

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

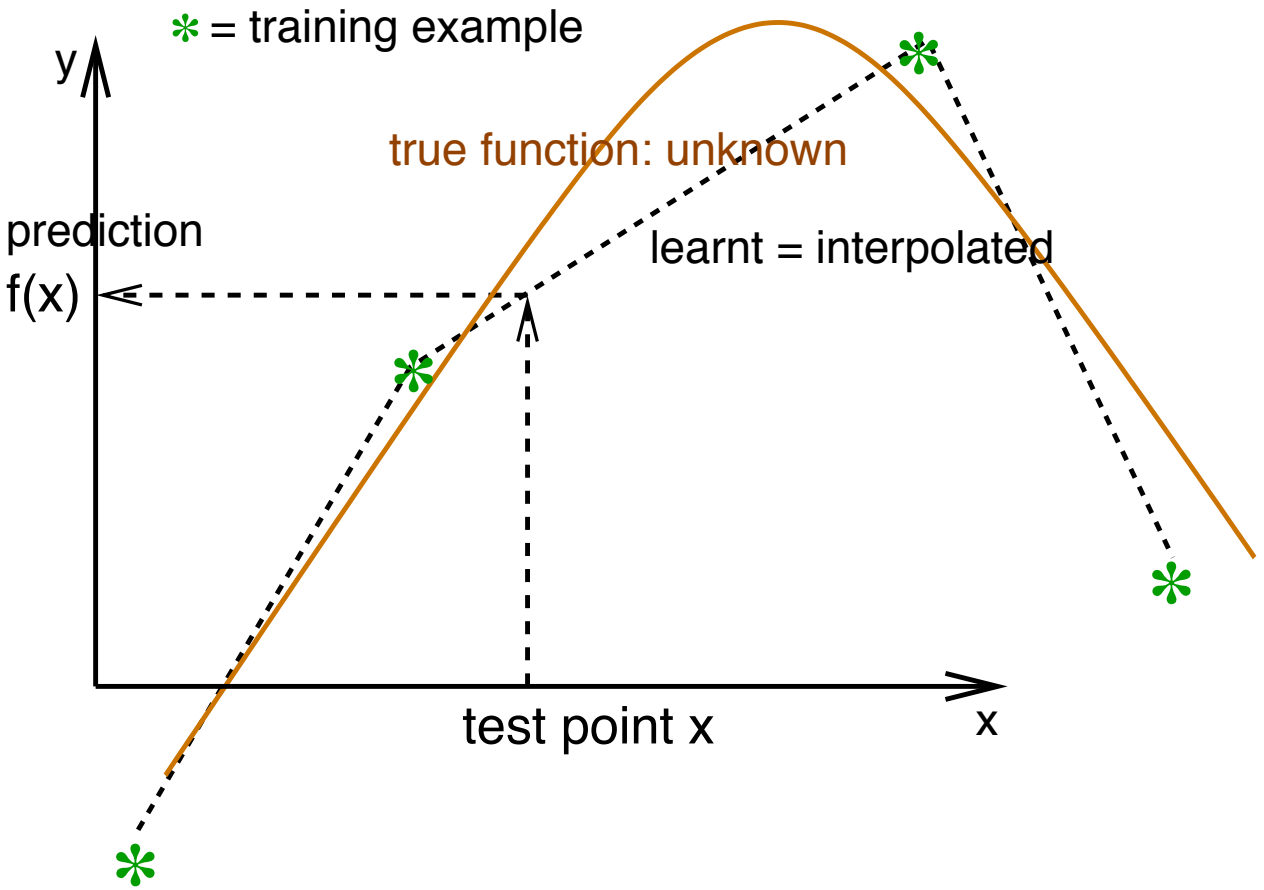
Classical solution: hope for a smooth enough target function, or make it smooth by handcrafting features



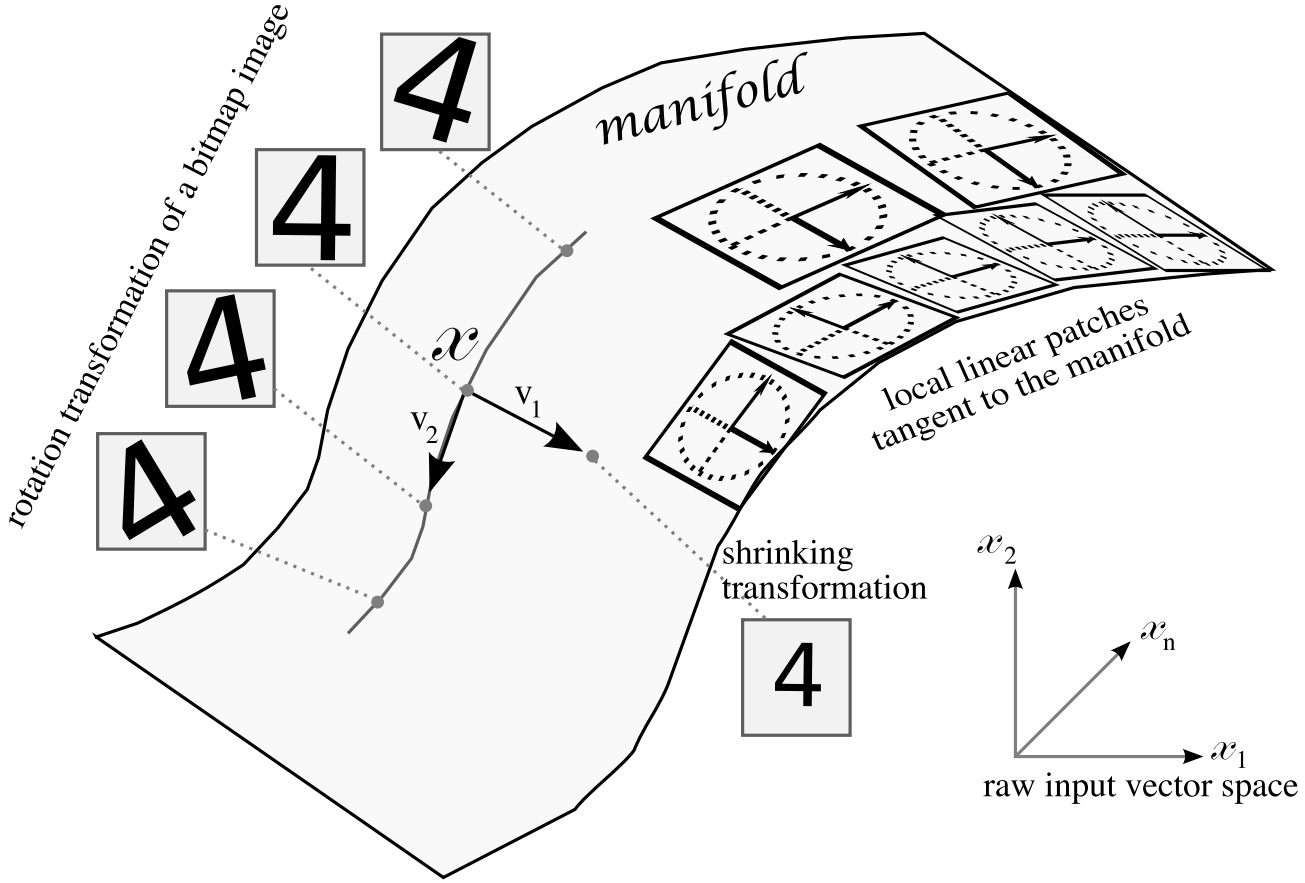
Easy Learning



Local Smoothness Prior: Locally Capture the Variations



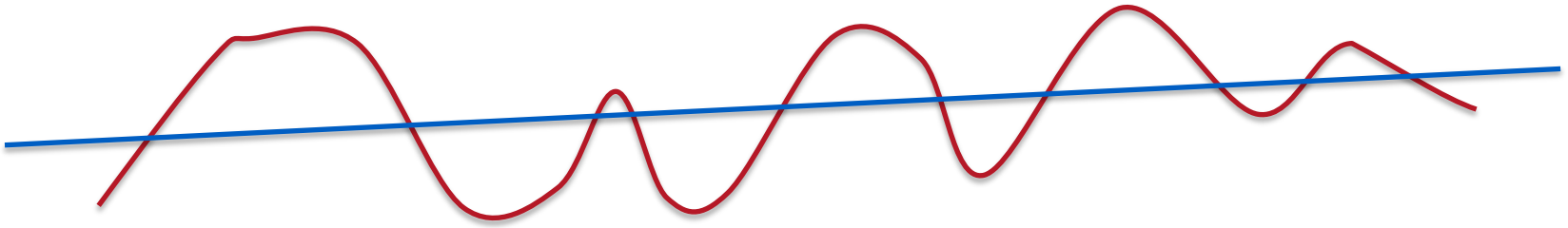
Real Data Are on Highly Curved Manifolds



Not Dimensionality so much as Number of Variations

(Bengio, Delalleau & Le Roux 2007)

- 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

Is there any hope to generalize non-locally?

Yes! Need more priors!

Part 1

Six Good Reasons to Explore Representation Learning

#1 Learning features, not just handcrafting them

Most ML systems use very carefully hand-designed features and representations

- Many practitioners are very experienced – and good – at such feature design (or kernel design)
- "Machine learning" often reduces to linear models (including CRFs) and nearest-neighbor-like features/models (including n-grams, kernel SVMs, etc.)

Hand-crafting features is time-consuming, brittle, incomplete

How can we automatically learn good features?

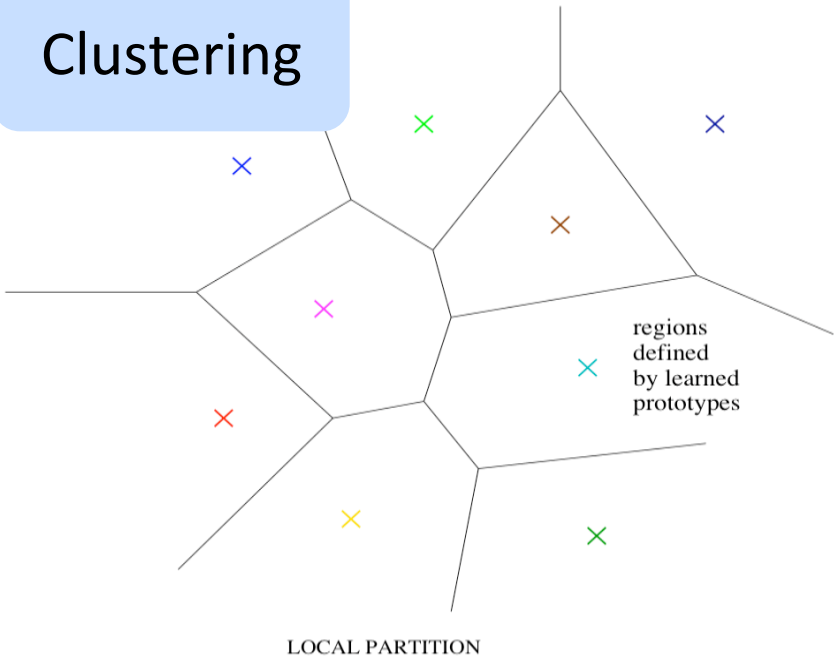
Claim: to approach AI, need to move scope of ML beyond hand-crafted features and simple models

Humans develop representations and abstractions to enable problem-solving and reasoning; our computers should do the same

Handcrafted features can be combined with learned features, or new more abstract features learned on top of handcrafted features

#2 The need for distributed representations

Clustering



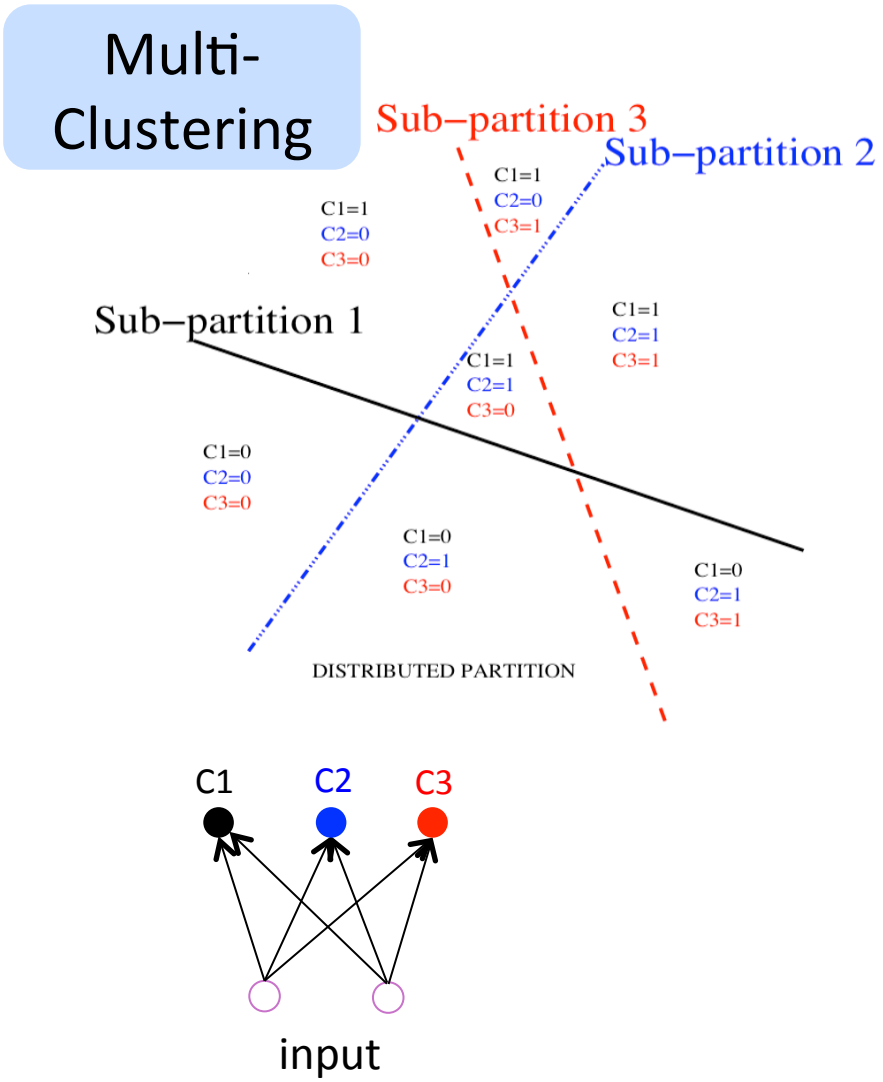
- Clustering, Nearest-Neighbors, RBF SVMs, local non-parametric density estimation & prediction, decision trees, etc.

- Parameters for each distinguishable region
- # distinguishable regions linear in # parameters

#2 The need for distributed representations

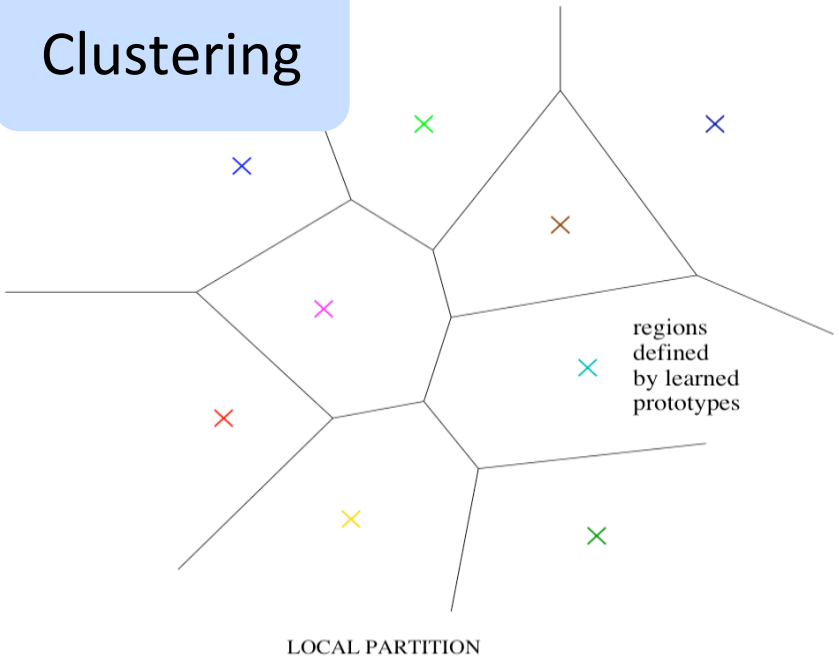
- Factor models, PCA, RBMs, Neural Nets, Sparse Coding, Deep Learning, etc.

- Each parameter influences many regions, not just local neighbors
- # distinguishable regions grows almost exponentially with # parameters
- **GENERALIZE NON-LOCALLY TO NEVER-SEEN REGIONS**

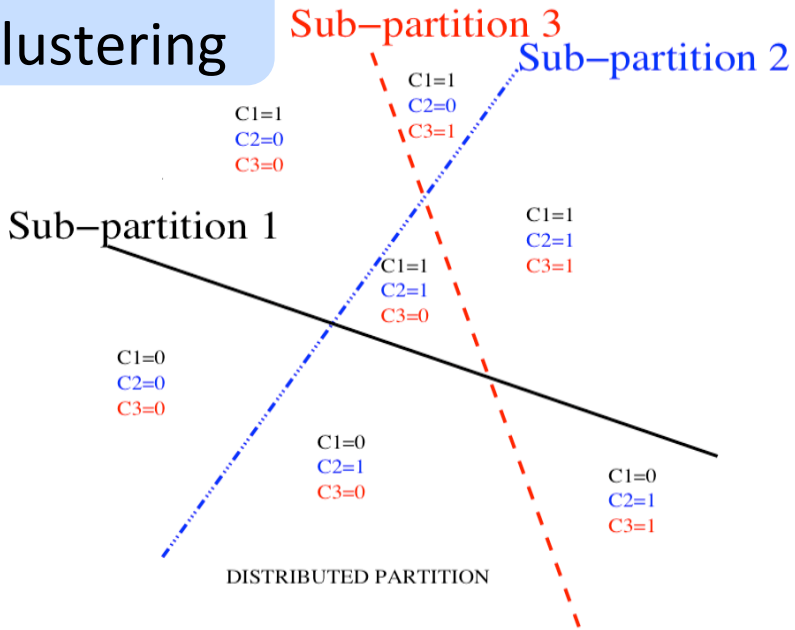


#2 The need for distributed representations

Clustering



Multi-Clustering



Learning a **set of features** that are not mutually exclusive can be exponentially more statistically efficient than nearest-neighbor-like or clustering-like models

#3 Unsupervised feature learning

Today, most practical ML applications require (lots of) labeled training data

- But almost all data is unlabeled

The brain needs to learn about 10^{14} synaptic strengths

- … in about 10^{9} seconds

Labels cannot possibly provide enough information

Most information acquired in an **unsupervised** fashion

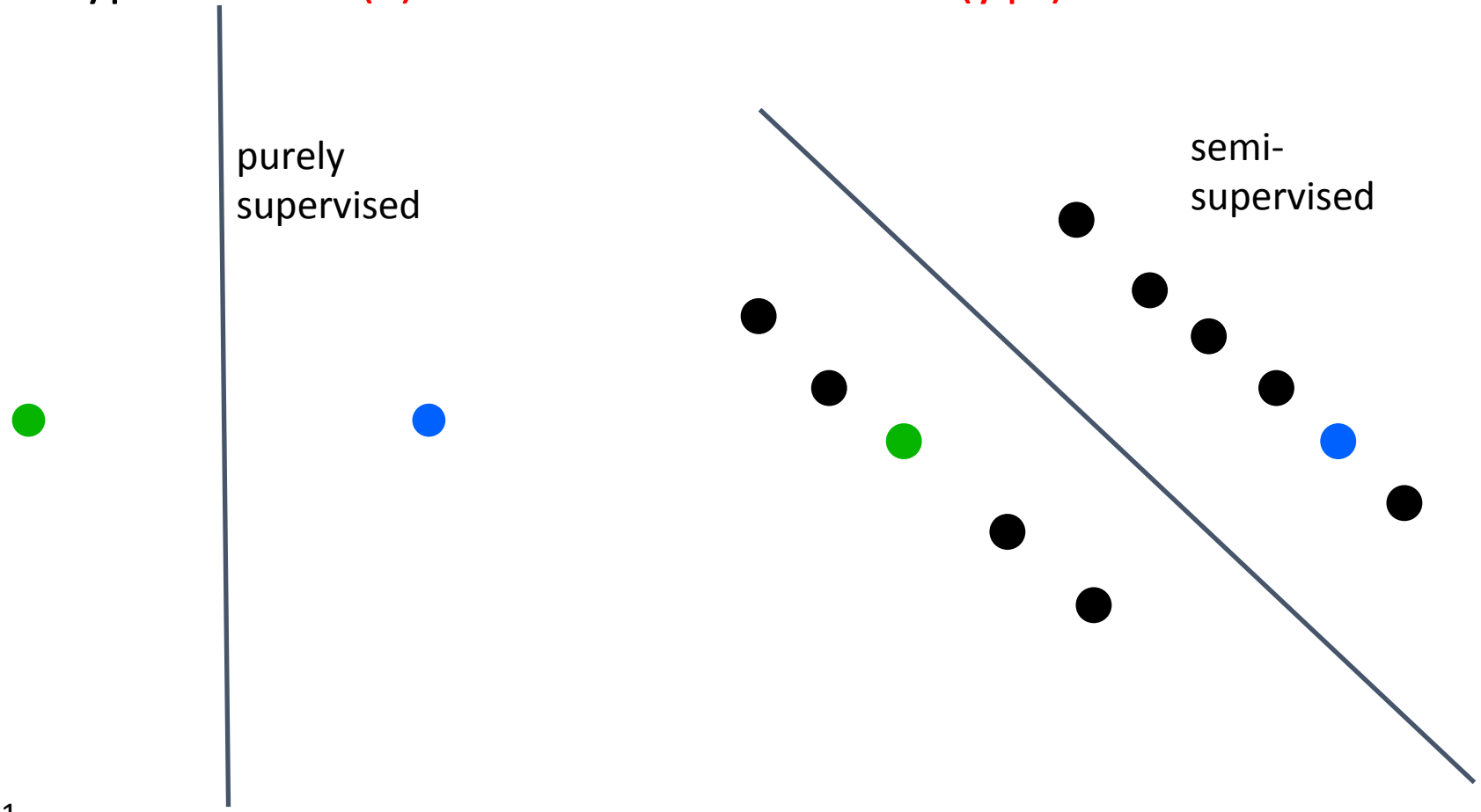
#3 How do humans generalize from very few examples?

- They **transfer** knowledge from previous learning:
  - Representations
  - Explanatory factors
- Previous learning from: unlabeled data

  \+ labels for other tasks
- **Prior: shared underlying explanatory factors, in particular between P(x) and P(Y|x)**

#3 Sharing Statistical Strength by Semi-Supervised Learning

- Hypothesis: P(x) shares structure with P(y|x)



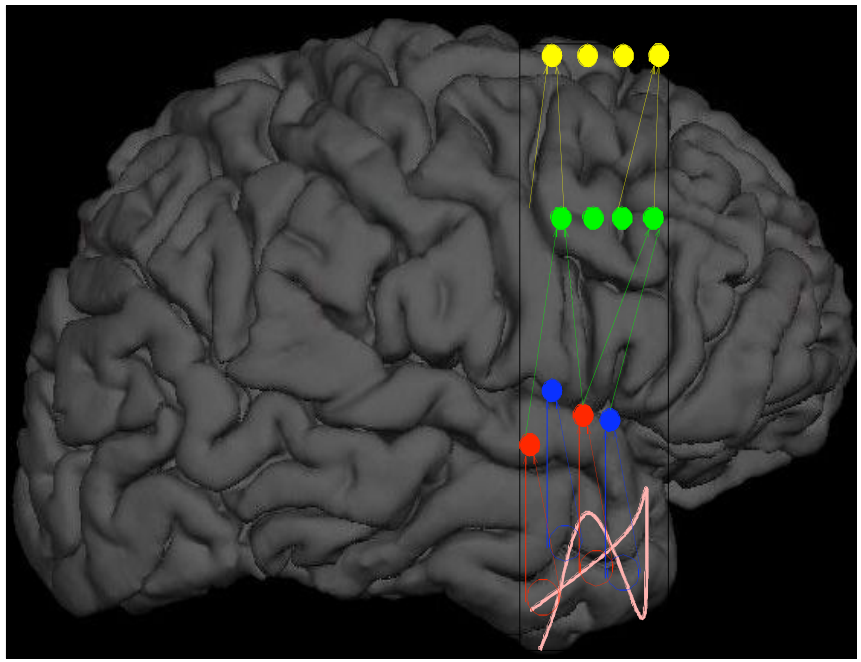
#4 Learning multiple levels of representation

There is theoretical and empirical evidence in favor of multiple levels of representation

Exponential gain for some families of functions

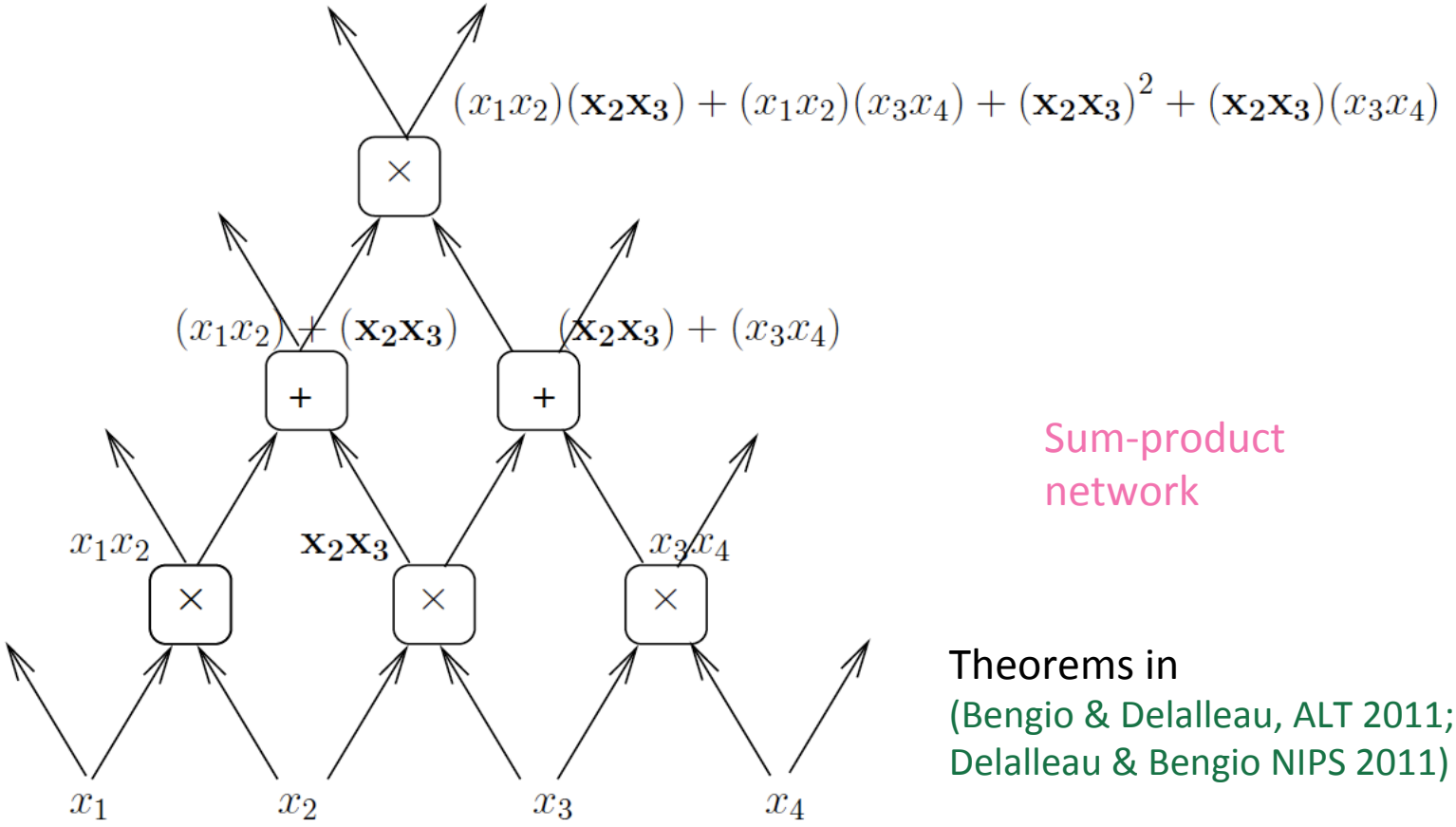
Biologically inspired learning

- Brain has a deep architecture
- Cortex seems to have a generic learning algorithm
- **Humans first learn simpler concepts and then compose them to more complex ones**



#4 Sharing Components in a Deep Architecture

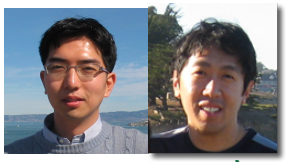
Polynomial expressed with shared components: advantage of depth may grow exponentially



Sum-product network

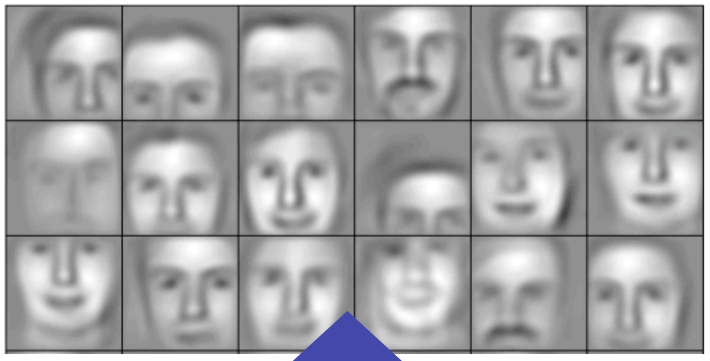
Theorems in
(Bengio & Delalleau, ALT 2011;
Delalleau & Bengio NIPS 2011)

#4 Learning multiple levels of representation



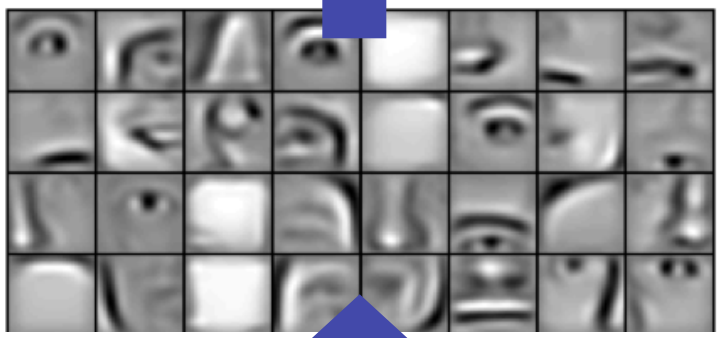
(Lee, Largman, Pham & Ng, NIPS 2009)
(Lee, Grosse, Ranganath & Ng, ICML 2009)

Successive model layers learn deeper intermediate representations



Layer 3

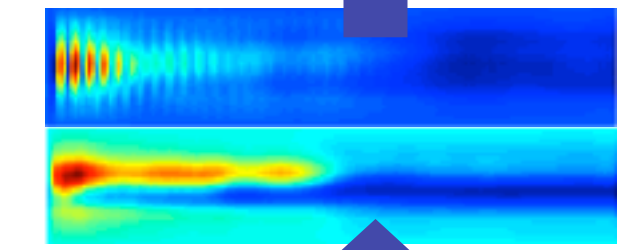
Parts combine to form objects

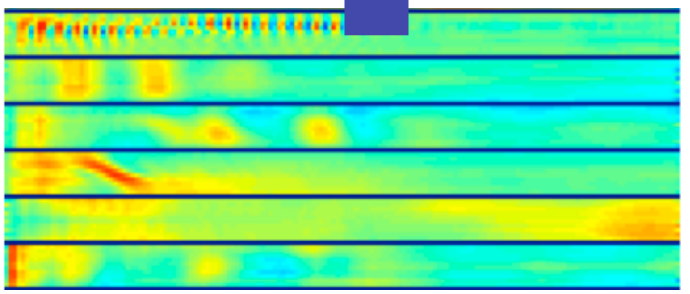


Layer 2

Layer 1

High-level linguistic representations

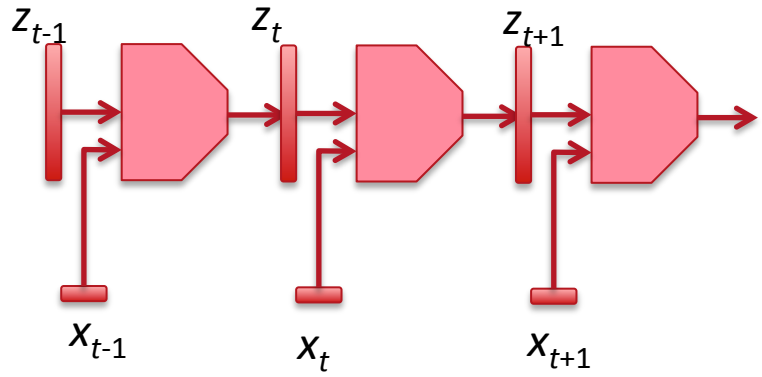


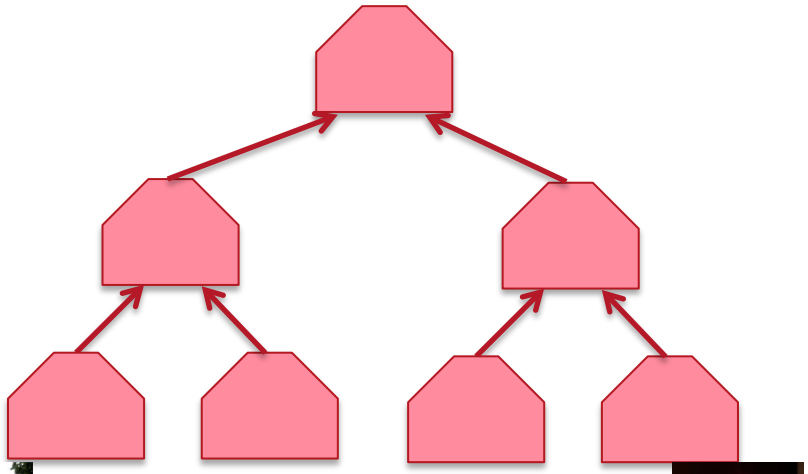


Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

#4 Handling the compositionality of human language and thought

- Human languages, ideas, and artifacts are composed from simpler components
- Recursion: the same operator (same parameters) is applied repeatedly on different states/components of the computation
- Result after unfolding = deep representations

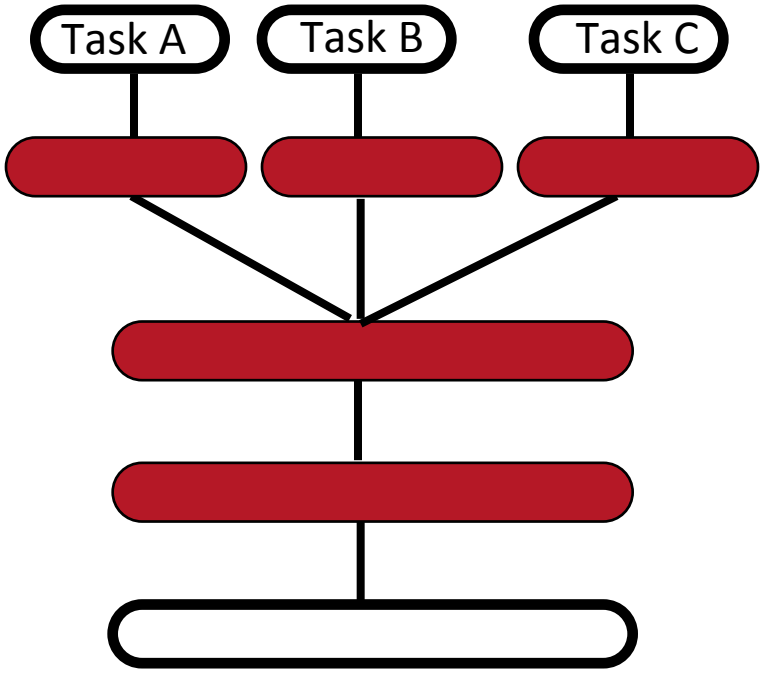




(Bottou 2011, Socher et al 2011)

#5 Multi-Task Learning

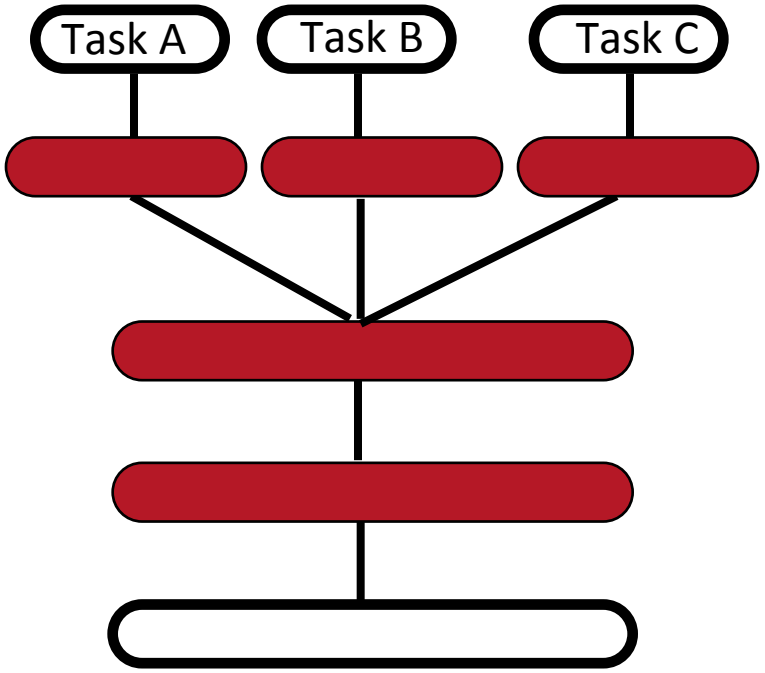
- Generalizing better to new tasks is crucial to approach AI
- Deep architectures learn good intermediate representations that can be shared across tasks
- Good representations that disentangle underlying factors of variation make sense for many tasks because each task concerns a subset of the factors



#5 Sharing Statistical Strength

- Multiple levels of latent variables also allow combinatorial sharing of statistical strength: intermediate levels can also be seen as sub-tasks

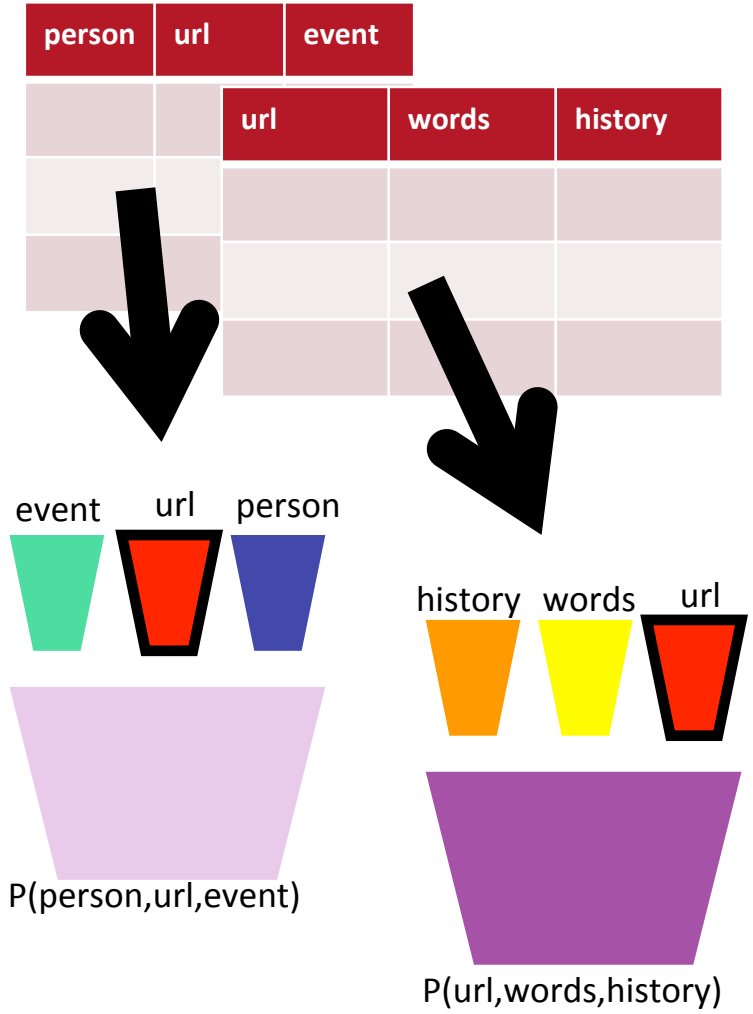
- E.g. dictionary, with intermediate concepts re-used across many definitions



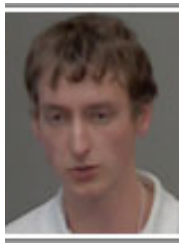
Prior: some shared underlying explanatory factors between tasks

#5 Combining Multiple Sources of Evidence with Shared Representations

- Traditional ML: data = matrix
- Relational learning: multiple sources, different tuples of variables
- Share representations of same types across data sources
- Shared learned representations help propagate information among data sources: e.g., WordNet, XWN, Wikipedia, FreeBase, ImageNet…
  (Bordes et al AISTATS 2012)



#5 Different object types represented in same space



Google:

S. Bengio, J. Weston & N. Usunier

(IJCAI 2011, NIPS'2010, JMLR 2010, MLJ 2010)

Learn $\Phi_I(\cdot)$ and $\Phi_W(\cdot)$ to optimize precision@k.

#6 Invariance and Disentangling

- Invariant features

- Which invariances?
- Alternative: learning to disentangle factors
- Good disentangling →

  avoid the curse of dimensionality

#6 Emergence of Disentangling

- (Goodfellow et al. 2009): sparse auto-encoders trained on images
  - some higher-level features more invariant to geometric factors of variation

- (Glorot et al. 2011): sparse rectified denoising auto-encoders trained on bags of words for sentiment analysis
  - different features specialize on different aspects (domain, sentiment)

WHY?

#6 Sparse Representations

- Just add a penalty on learned representation
- Information disentangling (compare to dense compression)
- More likely to be linearly separable (high-dimensional space)
- Locally low-dimensional representation = local chart
- Hi-dim. sparse = efficient variable size representation
  = data structure

Few bits of information

Many bits of information

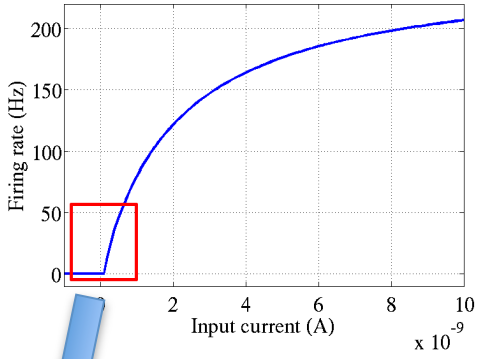
Prior: only few concepts and attributes relevant per example

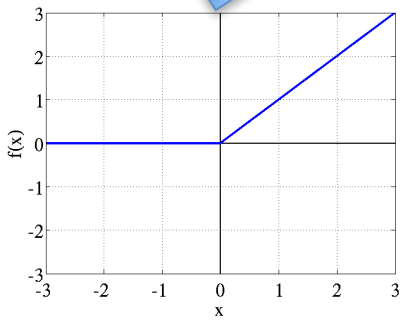
Deep Sparse Rectifier Neural Networks

(Glorot,Bordes and Bengio AISTATS 2011), following up on (Nair & Hinton 2010)

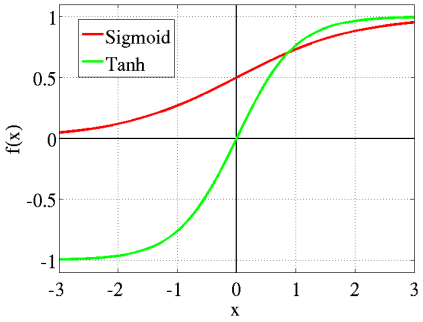
Neuroscience motivations

Leaky integrate-and-fire model





Rectifier



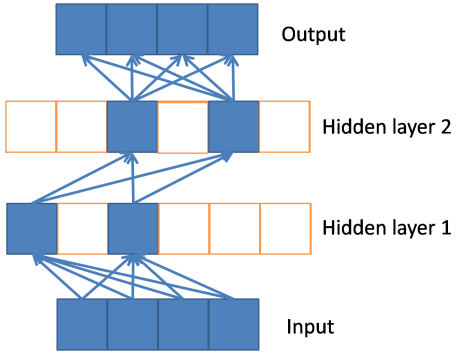
Commonly used functions

f(x)=max(0,x)

- *One-sided*
- Real zeros
- “default” regime at 0

Machine learning motivations

- Sparse representations
- Sparse and linear gradients



Deep Sparse Rectifier Neural Nets:

Can train deeper supervised nets!

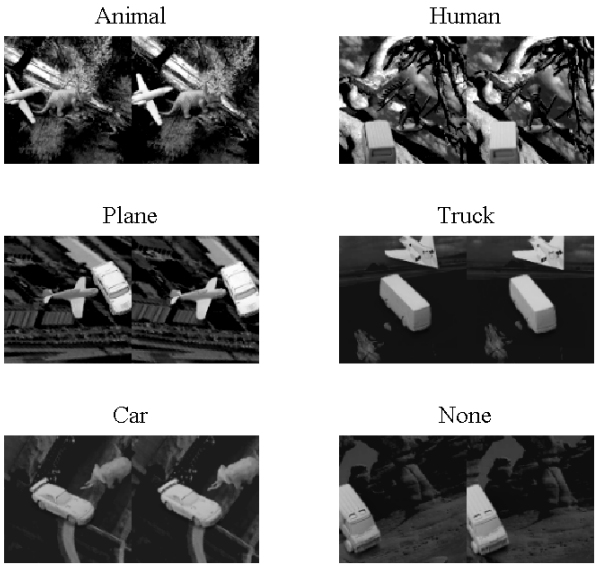
Experiments and results

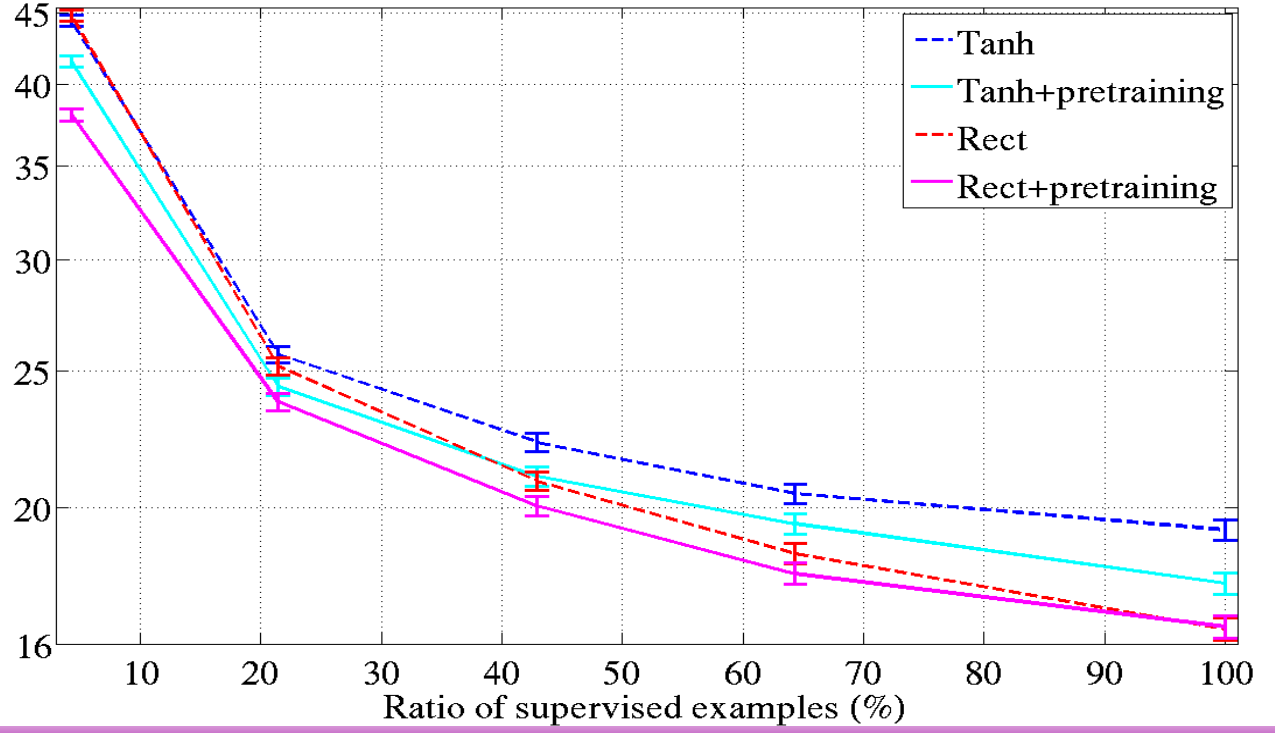
- Stacked denoising autoencoder
- 4 image recognition and 1 sentiment analysis datasets
- Better generalization than hyperbolic tangent networks
- Rectifier networks achieve their best performance without needing unsupervised pre-training
- Unsupervised pre-training is beneficial in the semi-supervised setting

Neuron	MNIST	CIFAR10	NISTP	NORB
With unsupervised pre-training				
Rectifier	**1.20%**	**49.96%**	**32.86%**	**16.46%**
Tanh	**1.16%**	**50.79%**	35.89%	17.66%
Softplus	**1.17%**	**49.52%**	**33.27%**	19.19%
Without unsupervised pre-training				
Rectifier	**1.43%**	**50.86%**	**32.64%**	**16.40%**
Tanh	1.57%	52.62%	36.46%	19.29%
Softplus	1.77%	53.20%	35.48%	17.68%

NISTP

NORB





Temporal Coherence and Scales

- One of the hints from nature about different explanatory factors:
  - Rapidly changing factors (often noise)
  - Slowly changing (generally more abstract)
  - Different factors at different time scales

- We should exploit those **hints** to **disentangle** better!
- (Becker & Hinton 1993, Wiskott & Sejnowski 2002, Hurri & Hyvarinen 2003, Berkes & Wiskott 2005, Mobahi et al 2009, Bergstra & Bengio 2009)

Bypassing the curse

We need to build compositionality into our ML models

Just as human languages exploit compositionality to give representations and meanings to complex ideas

Exploiting compositionality gives an exponential gain in representational power

Distributed representations / embeddings: feature learning

Deep architecture: multiple levels of feature learning

Prior: compositionality is useful to describe the world around us efficiently

Bypassing the curse by sharing statistical strength

- Besides very fast GPU-enabled predictors, the main advantage of representation learning is statistical: potential to learn from less labeled examples because of sharing of statistical strength:
  - Unsupervised pre-training and semi-supervised training
  - Multi-task learning
  - Multi-data sharing, learning about symbolic objects and their relations

Why now?

Despite prior investigation and understanding of many of the algorithmic techniques …

Before 2006 training deep architectures was **unsuccessful**

(except for convolutional neural nets when used by people who speak French)

What has changed?

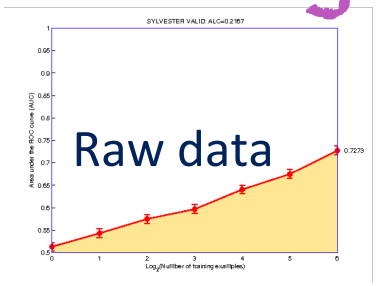
- New methods for unsupervised pre-training have been developed (variants of Restricted Boltzmann Machines = RBMs, regularized autoencoders, sparse coding, etc.)
- Better understanding of these methods
- Successful real-world applications, winning challenges and beating SOTAs in various areas

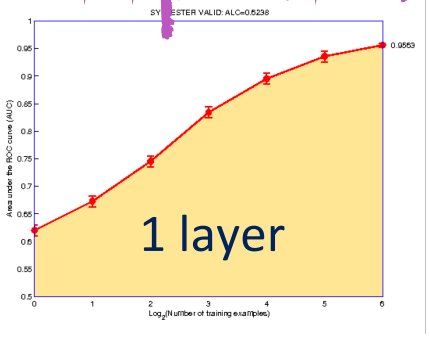
Major Breakthrough in 2006

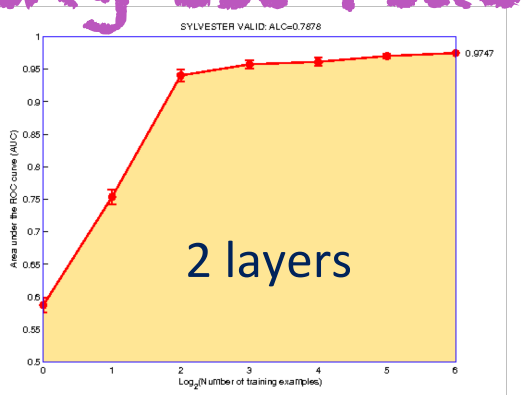
- Ability to train deep architectures by using layer-wise unsupervised learning, whereas previous purely supervised attempts had failed

- Unsupervised feature learners:
  - RBMs
  - Auto-encoder variants
  - Sparse coding variants

Unsupervised and Transfer Learning Challenge + Transfer Learning Challenge: Deep Learning 1st Place

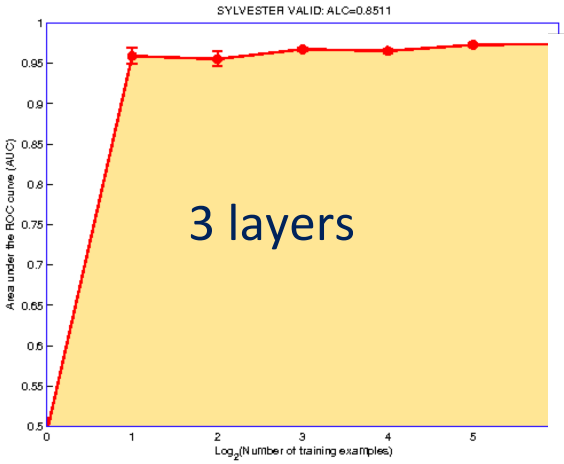


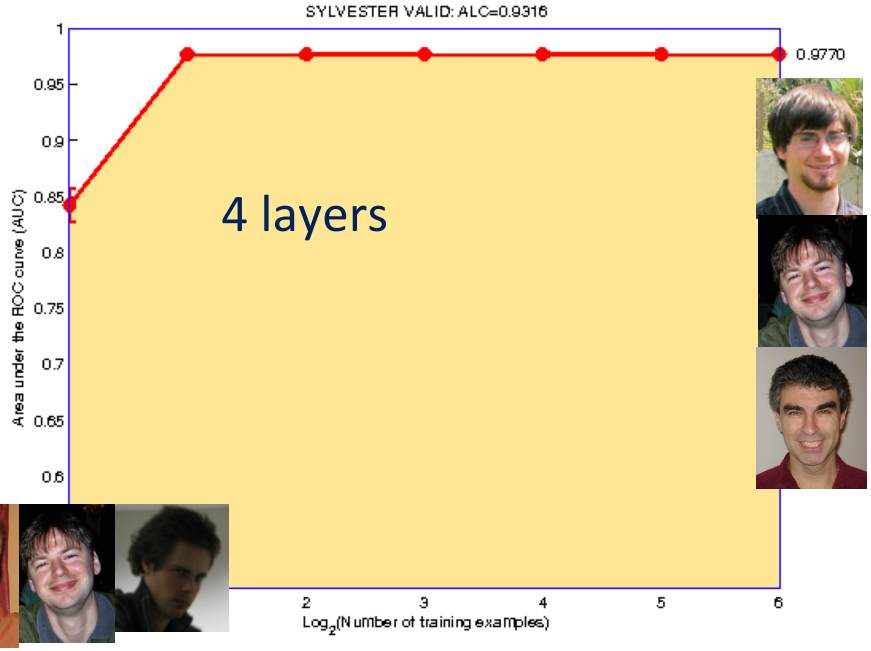




NIPS'2011 Transfer Learning Challenge Paper: ICML'2012

ICML'2011 workshop on Unsup. & Transfer Learning



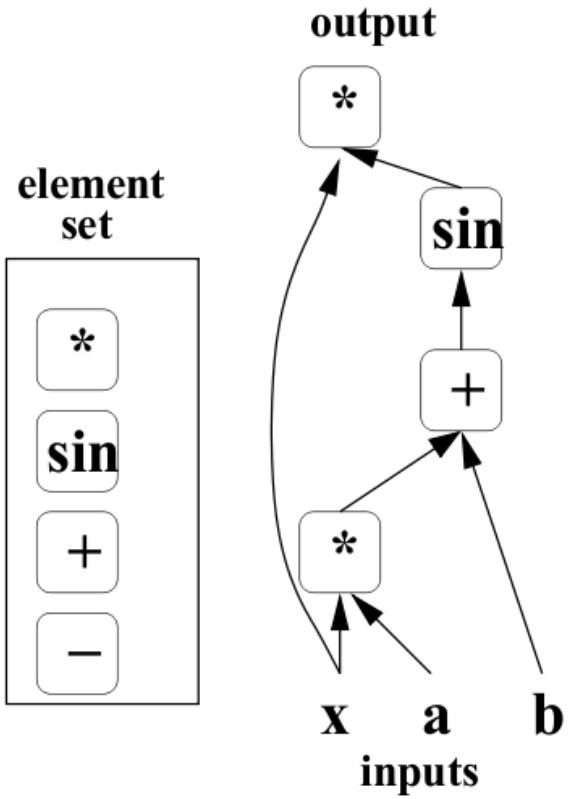


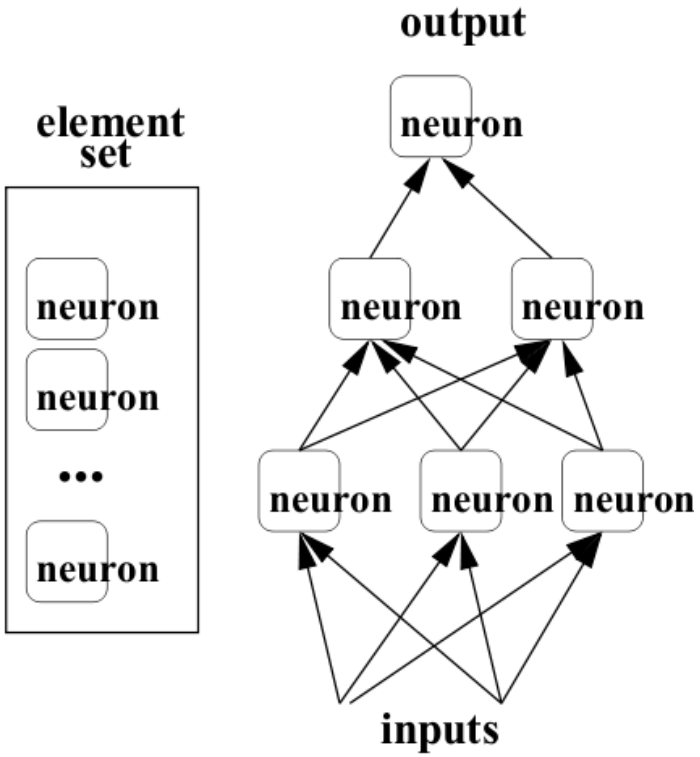
More Successful Applications

- Microsoft uses DL for speech rec. service (audio video indexing), based on Hinton/Toronto's DBNs (Mohamed et al 2011)
- Google uses DL in its Google Goggles service, using Ng/Stanford DL systems
- NYT talks about these: http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html?_r=1
- Substantially beating SOTA in language modeling (perplexity from 140 to 102 on Broadcast News) for speech recognition (WSJ WER from 16.9% to 14.4%) (Mikolov et al 2011) and translation (+1.8 BLEU) (Schwenk 2012)
- SENNA: Unsup. pre-training + multi-task DL reaches SOTA on POS, NER, SRL, chunking, parsing, with >10x better speed & memory (Collobert et al 2011)
- Recursive nets surpass SOTA in paraphrasing (Socher et al 2011)
- Denoising AEs substantially beat SOTA in sentiment analysis (Glorot et al 2011)
- Contractive AEs SOTA in knowledge-free MNIST (.8% err) (Rifai et al NIPS 2011)
- Le Cun/NYU's stacked PSDs most accurate & fastest in pedestrian detection and DL in top 2 winning entries of German road sign recognition competition

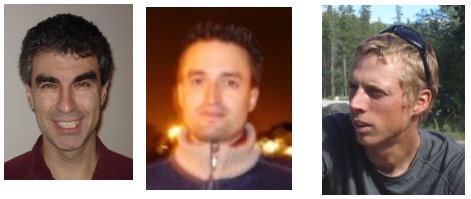
More about depth

Architecture Depth





Deep Architectures are More Expressive



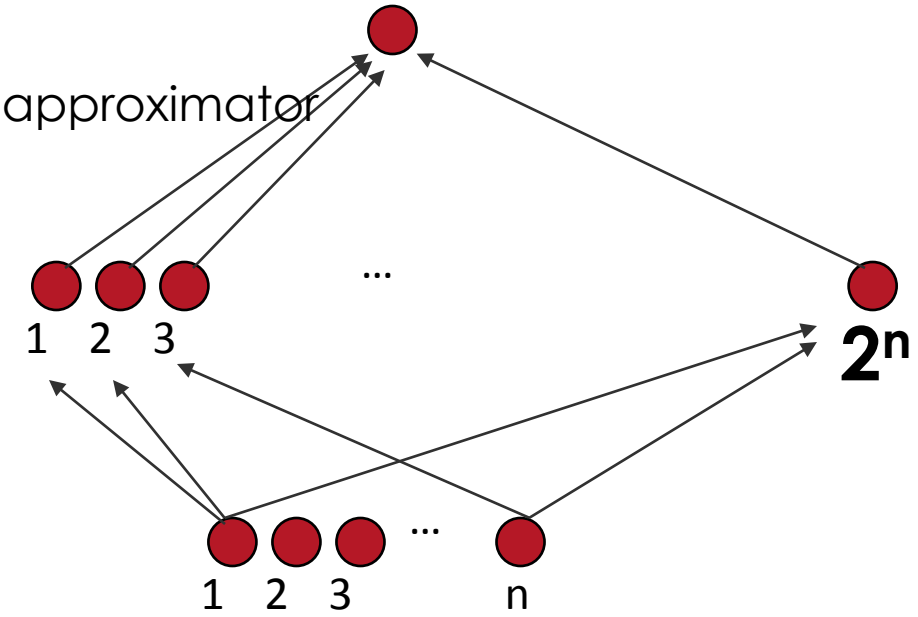
Theoretical arguments:

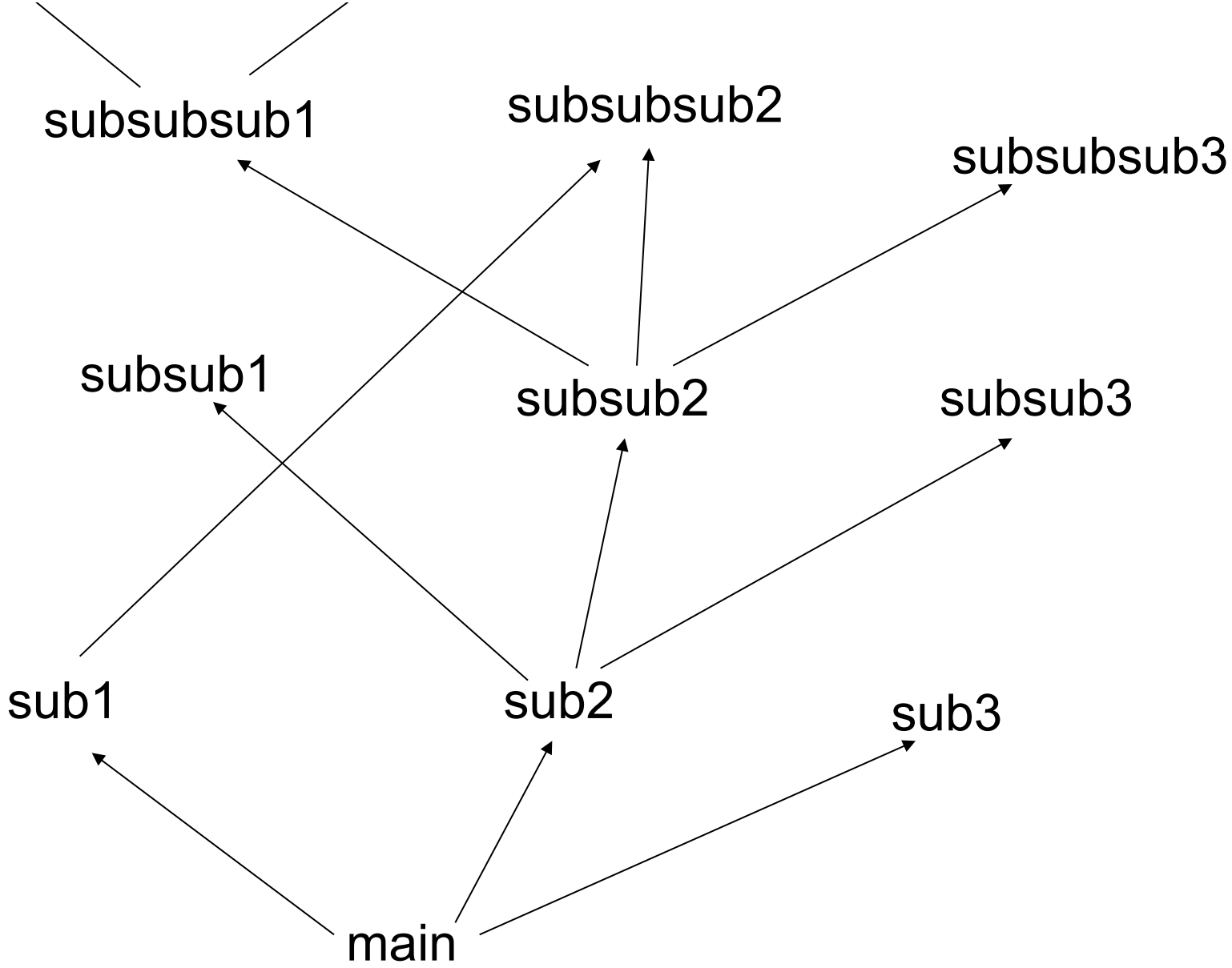
2 layers of { Logic gates / Formal neurons / RBF units } = universal approximator

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 2 layers

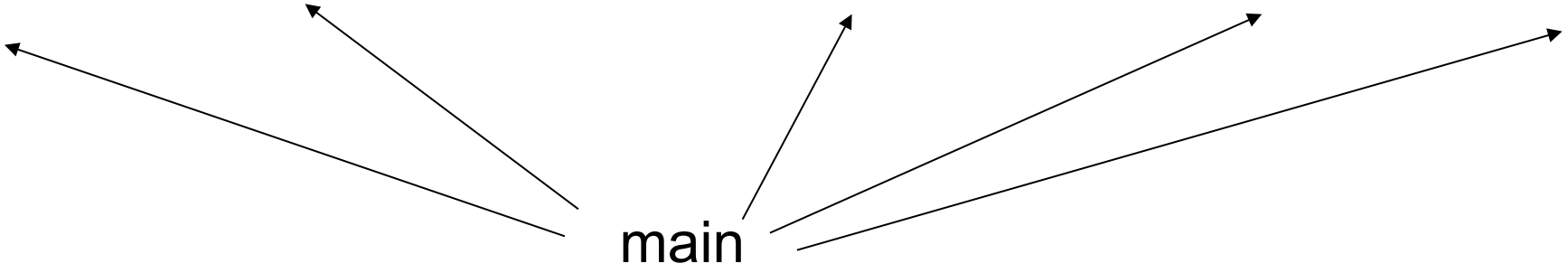




“Deep” computer program

subroutine1 includes
subsub1 code and
subsub2 code and
subsubsub1 code

subroutine2 includes
subsub2 code and
subsub3 code and
subsubsub3 code and …



“Shallow” computer program

“Deep” circuit

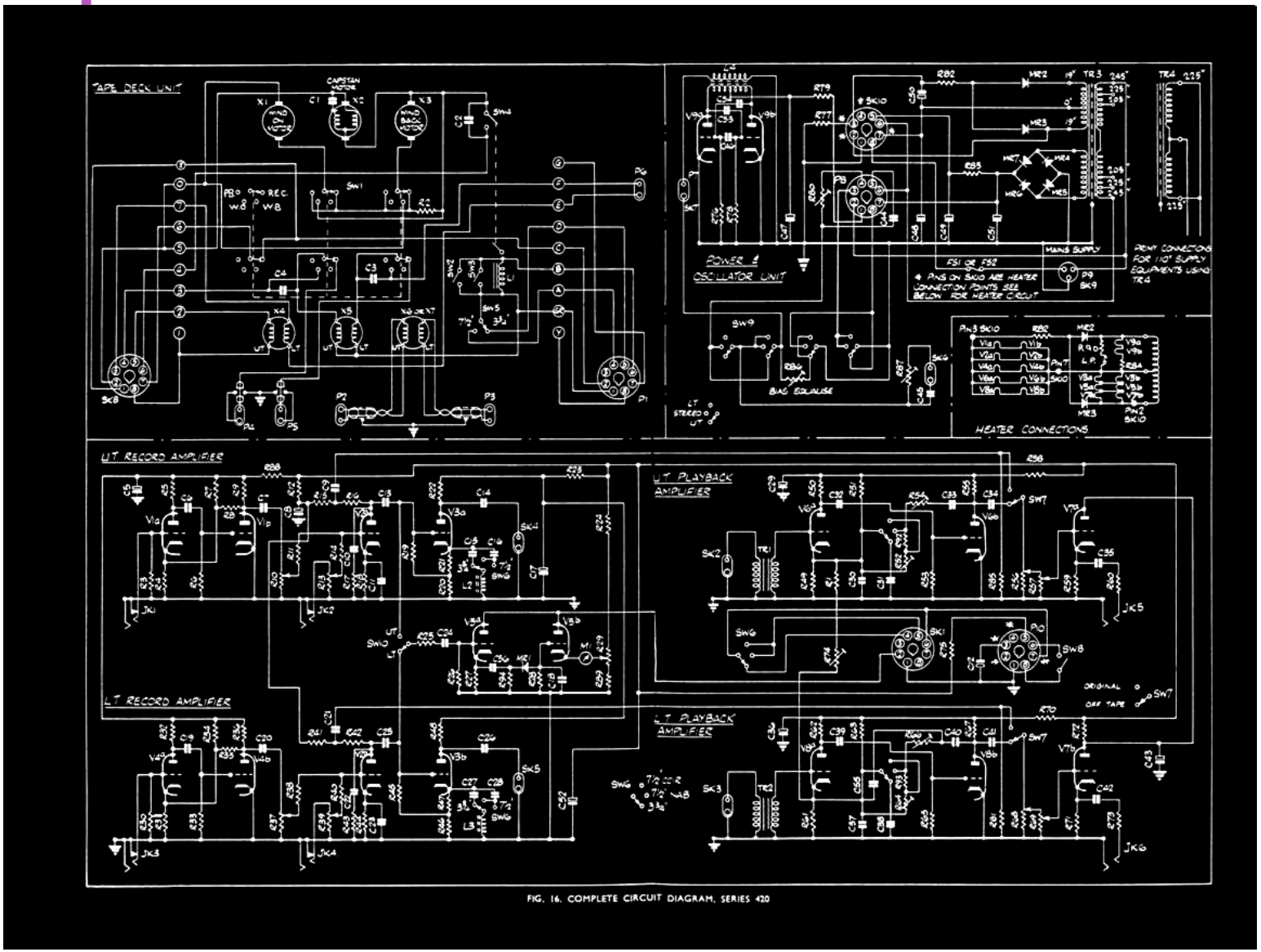
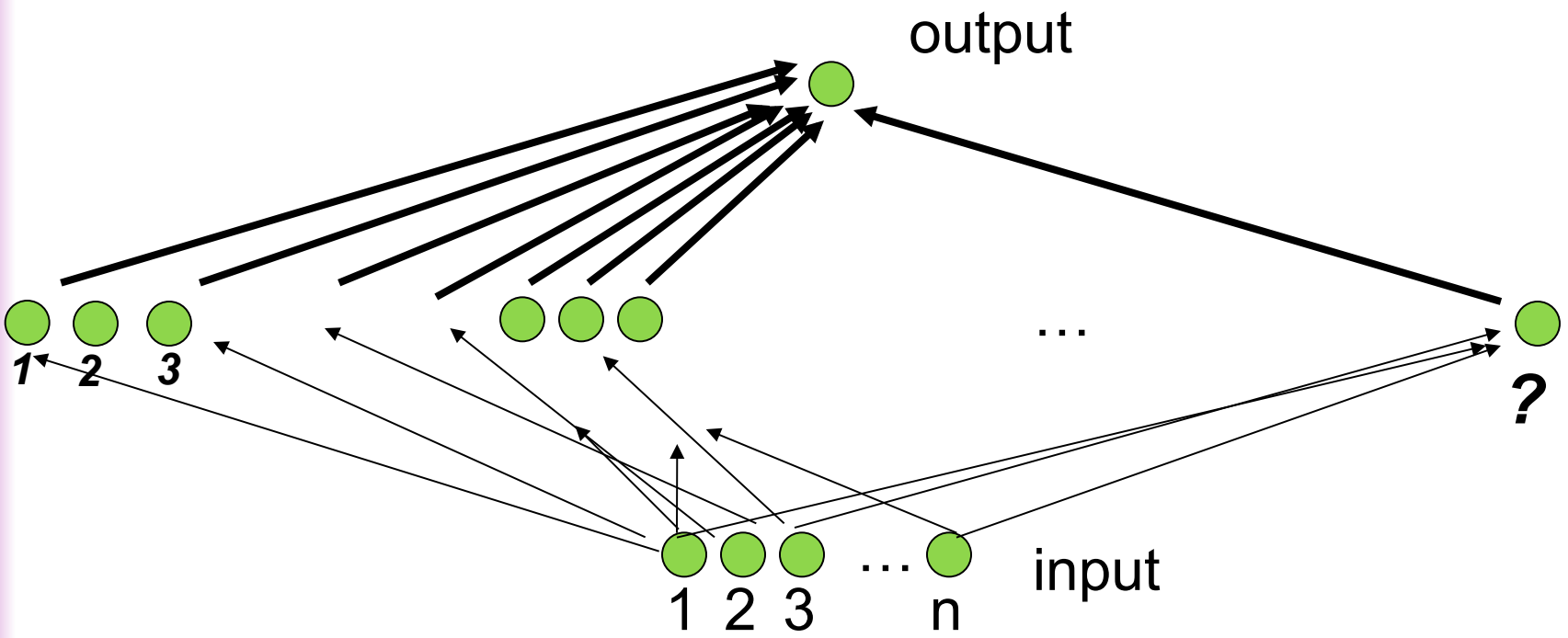


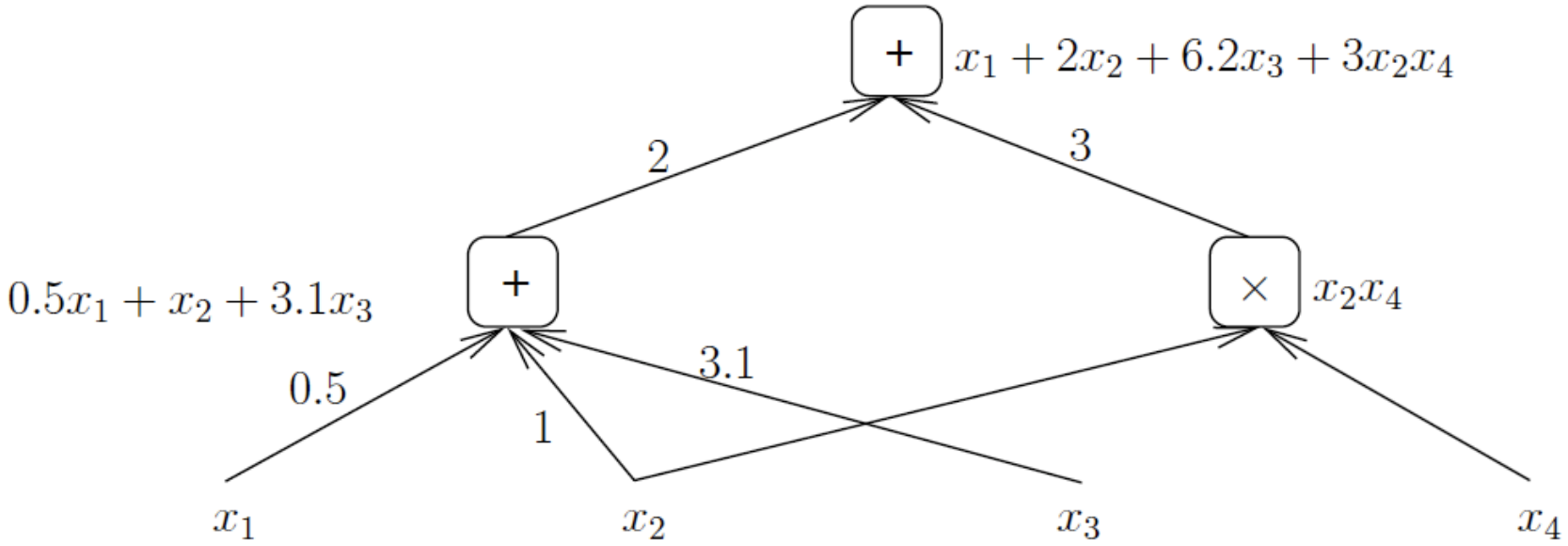
FIG. 16. COMPLETE CIRCUIT DIAGRAM, SERIES 420

"Shallow" circuit



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

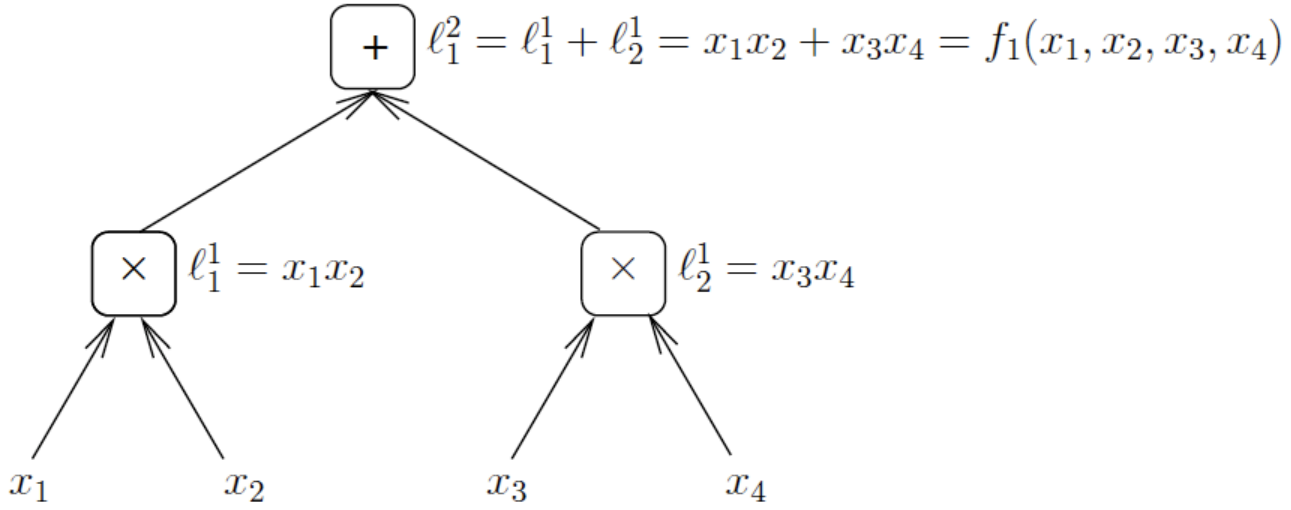
Sum-Product Networks



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

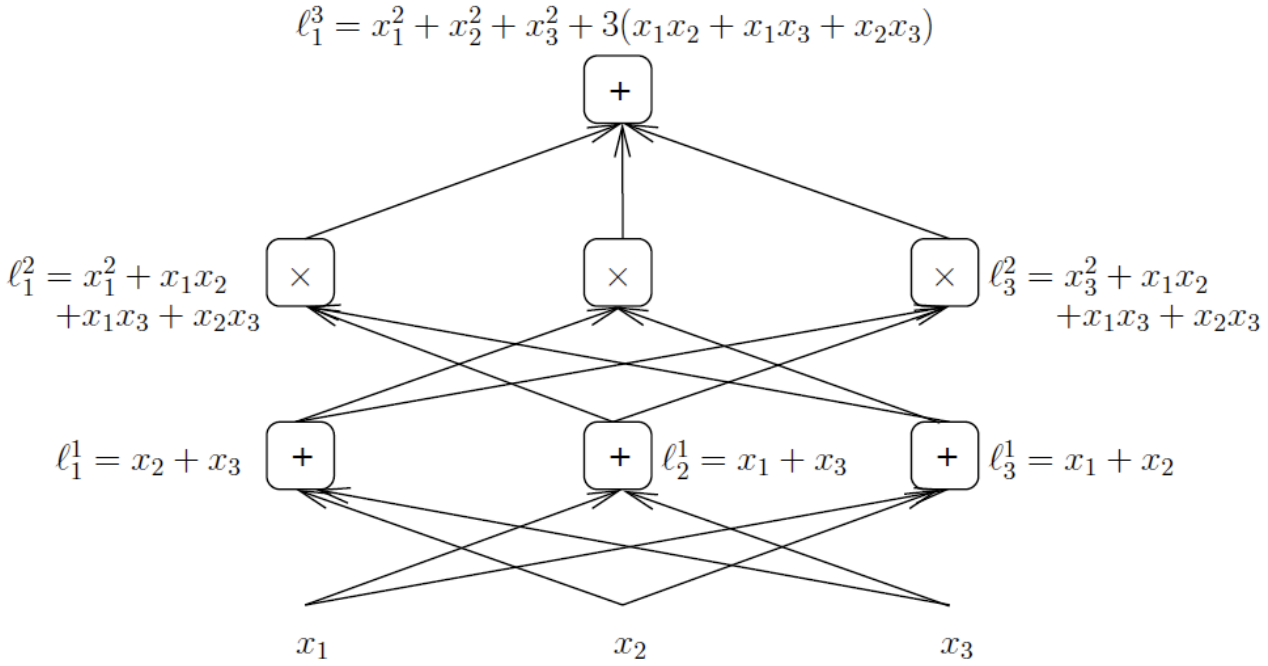


- $2i$ layers and $n = 4^i$ input variables
- alternate additive and multiplicative units
- unit ℓ_j^k takes as inputs ℓ_{2j-1}^{k-1} and ℓ_{2j}^{k-1}

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 $\{\ell_m^{k-1} | m \neq j\}$

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.