# Deep Learning of Representations

**Yoshua Bengio**

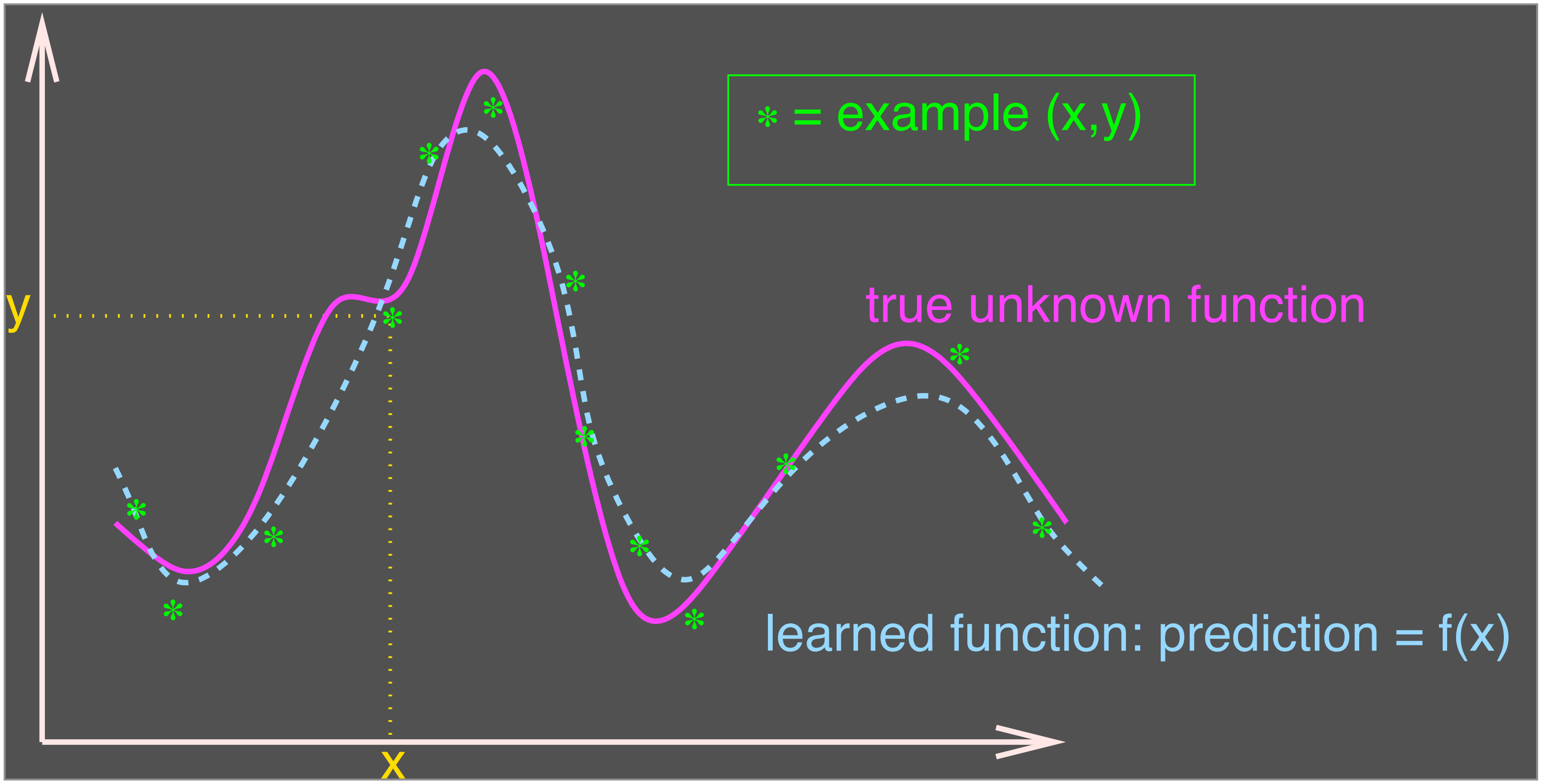
Département d'Informatique et Recherche Opérationnelle, U. Montréal

22 novembre 2012, Google Montreal

# Ultimate Goals

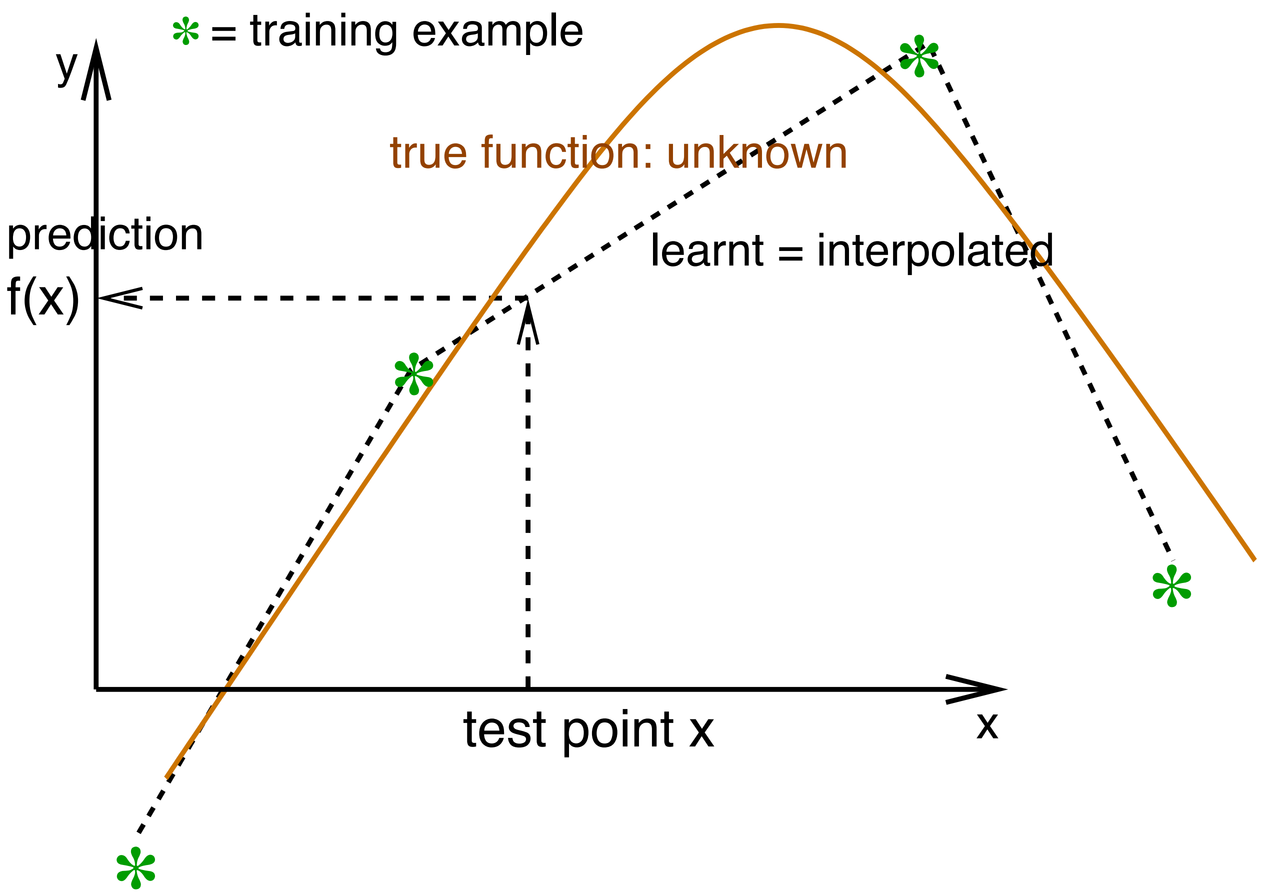
- AI
- Needs knowledge
- Needs **learning**
- Needs generalizing where probability mass concentrates
- Needs to fight the curse of dimensionality
- Needs disentangling the underlying explanatory factors ("making sense of the data")

# Easy Learning



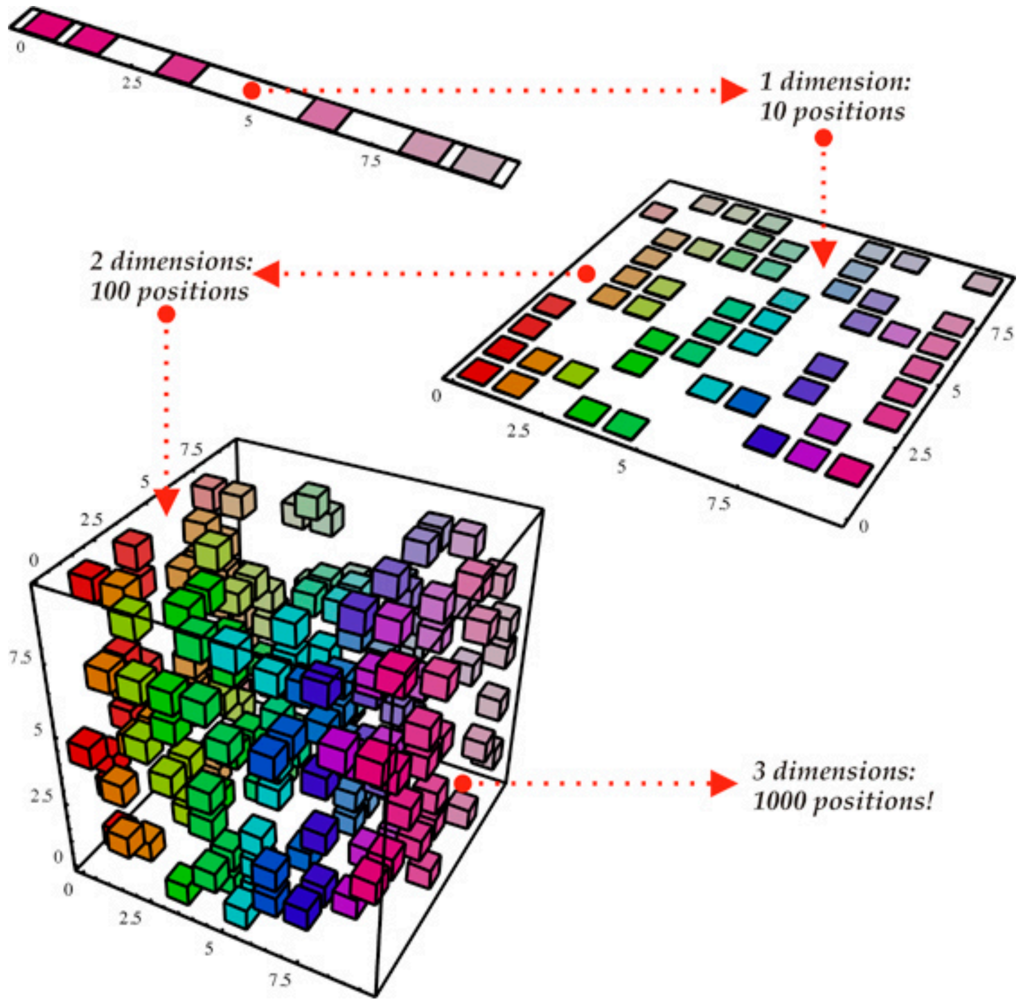
# Local Smoothness Prior: Locally Capture the Variations

$x \approx x' \Rightarrow f(x) \approx f(x')$



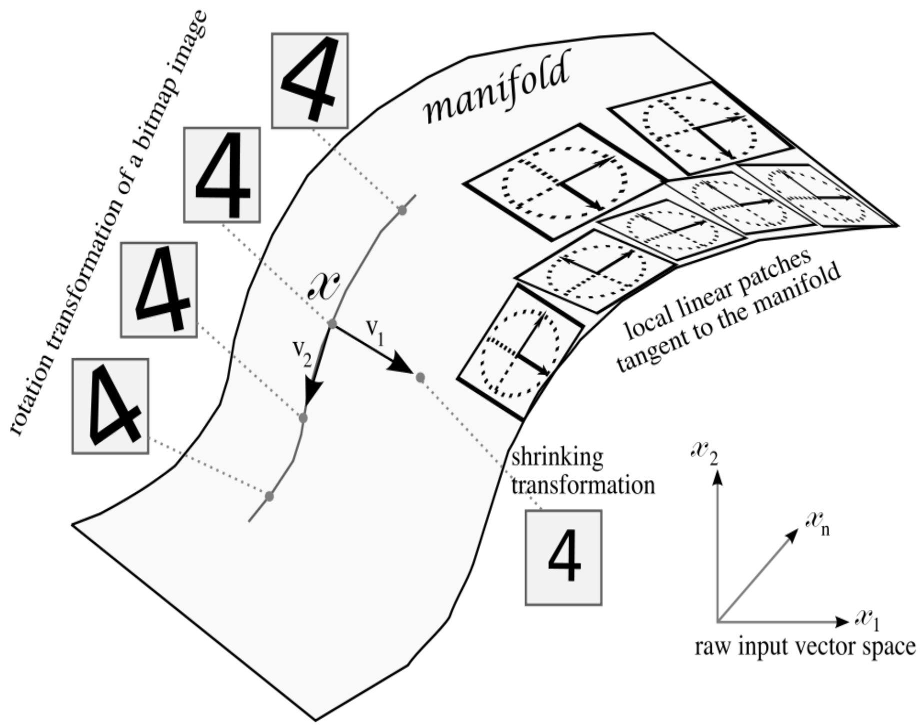
# What We Are Fighting Against: The Curse of Dimensionality

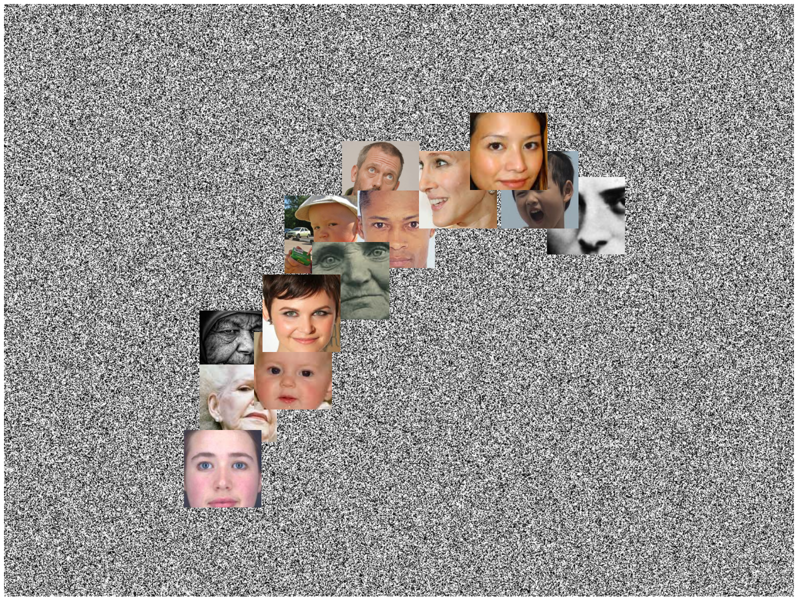
To generalize locally, need representative examples for all relevant variations!



# Manifold Learning

Prior: examples **concentrate** near lower dimensional manifold

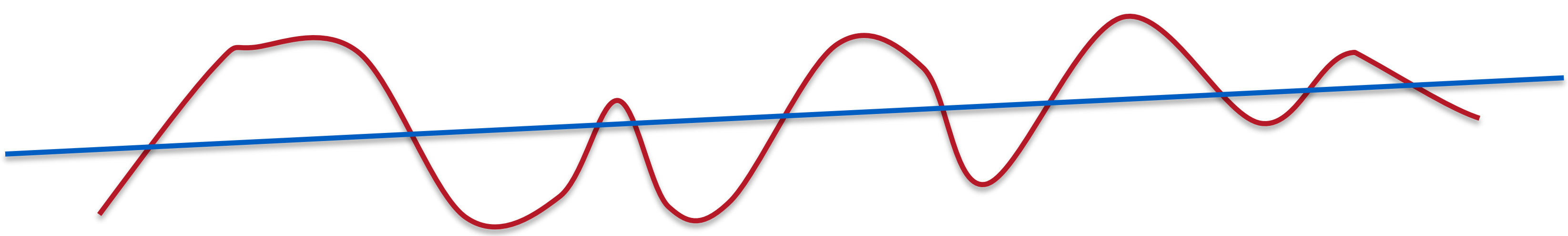


# Not Dimensionality so much as Number of Variations

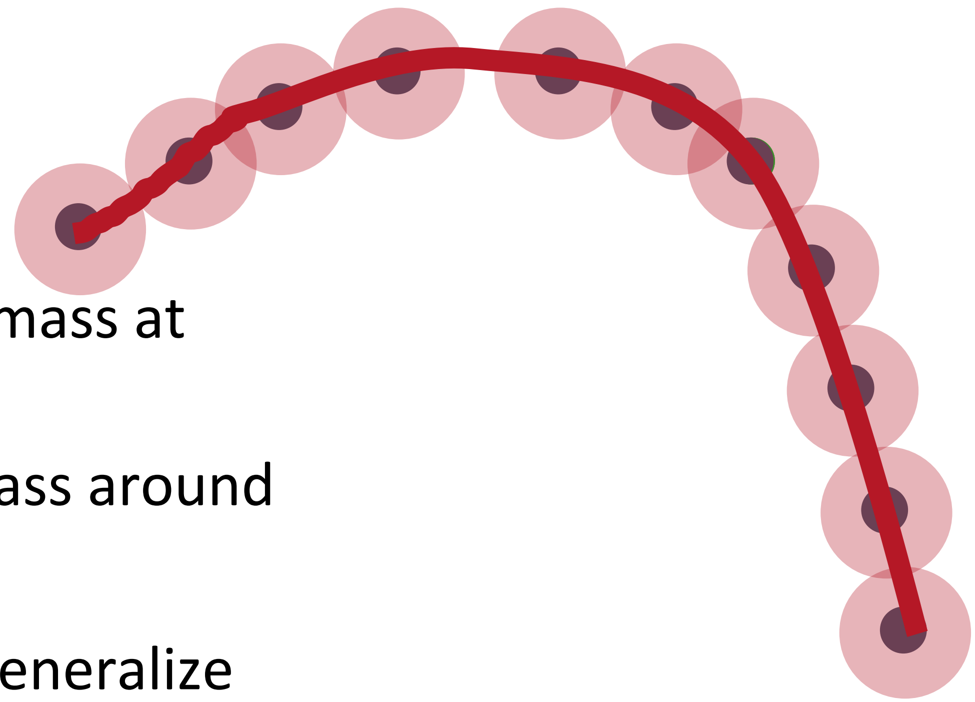
(Bengio, Delalleau & Le Roux 2007)

- 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

# Putting Probability Mass where Structure is Plausible

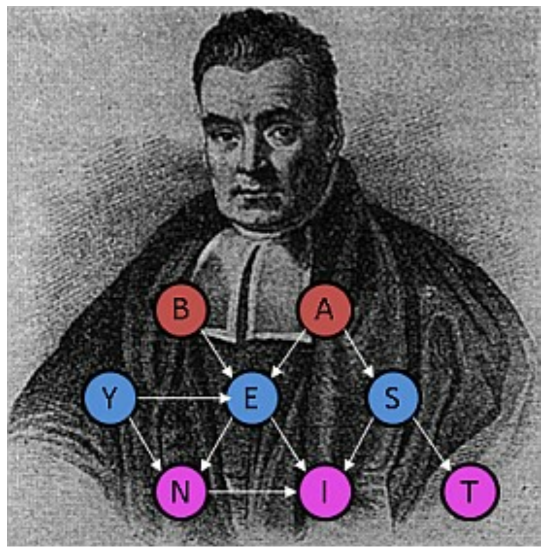


- Empirical distribution: mass at training examples
- Smoothness: spread mass around
- Insufficient
- Guess 'structure' and generalize accordingly

# Representation Learning

- Good input features essential for successful ML
  *(feature engineering = 90% of effort in industrial ML)*
- Handcrafting features vs learning them

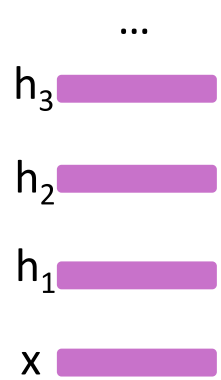
- Representation learning: guesses
  the features / factors / causes =
  good representation.



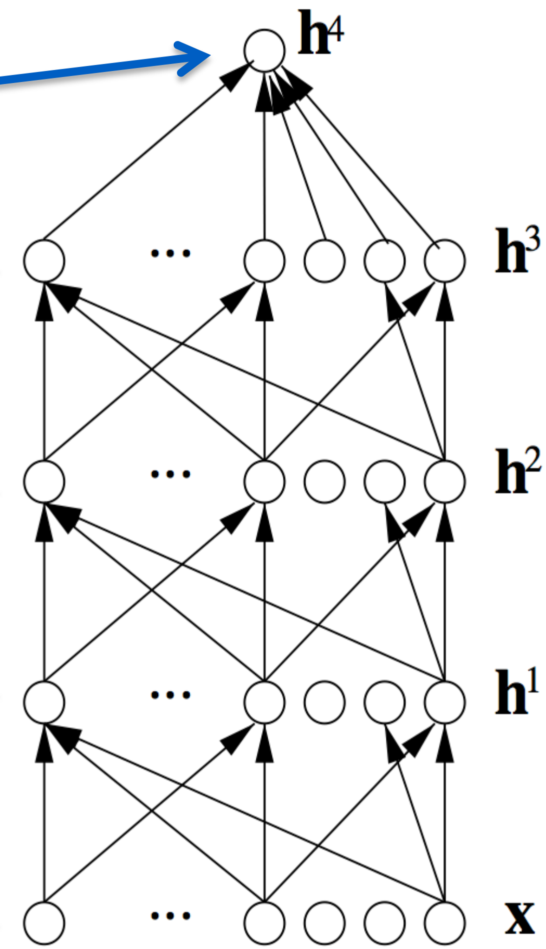
# Deep Representation Learning

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* *Deep Learning*



# A Good Old Deep Architecture




Optional Output layer

Here predicting a supervised target

Hidden layers

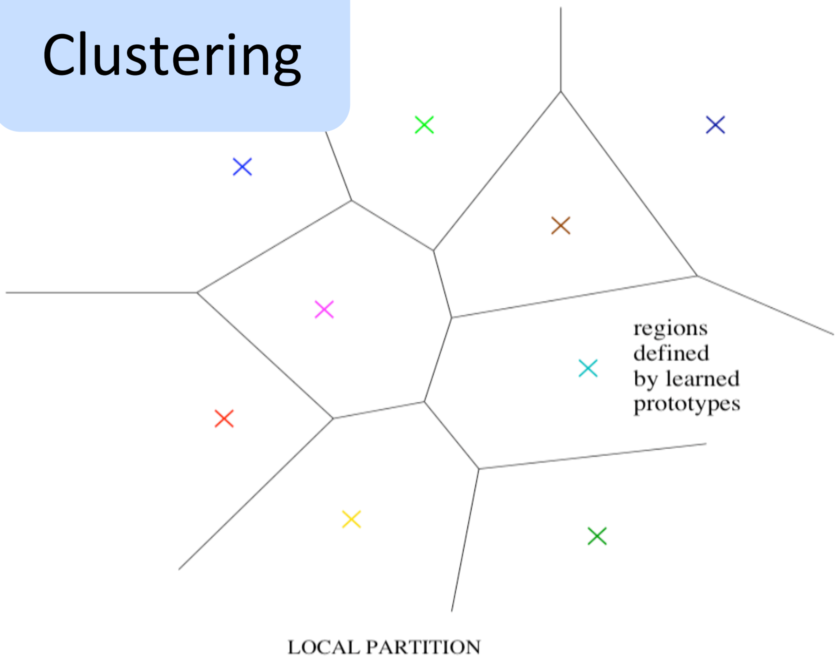
These learn more abstract representations as you head up

Input layer

This has raw sensory inputs (roughly)

# Generalizing Locally

Clustering

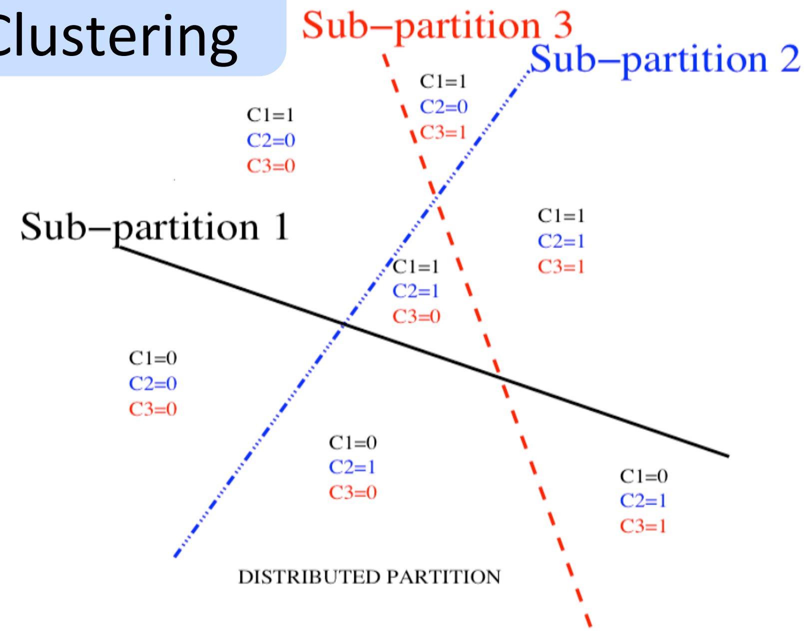


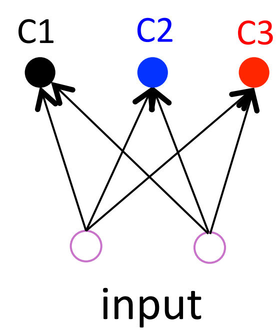

- Clustering, Nearest-Neighbors, RBF SVMs, local non-parametric density estimation & prediction, decision trees, etc.
- Parameters for each distinguishable region
- # distinguishable regions linear in # parameters

# 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**

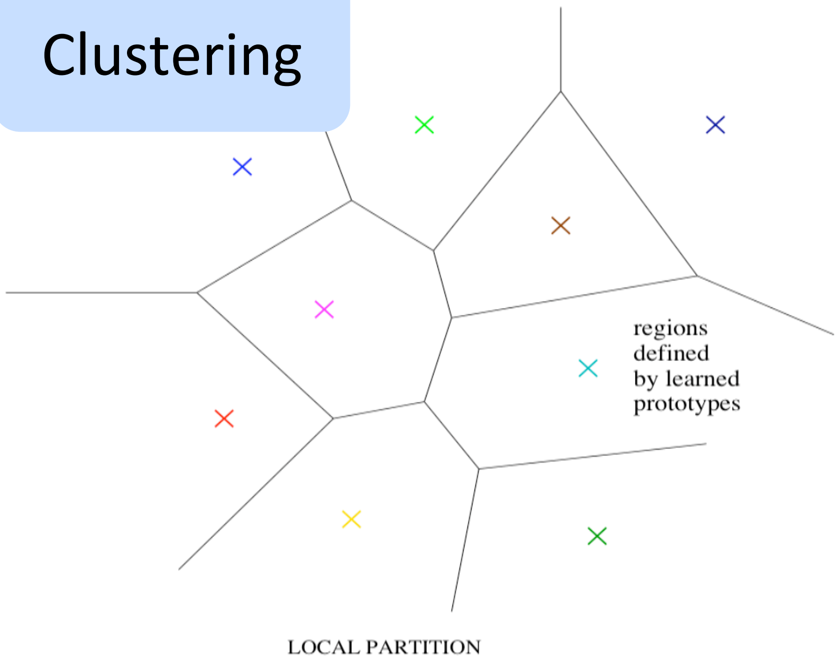
Multi-Clustering



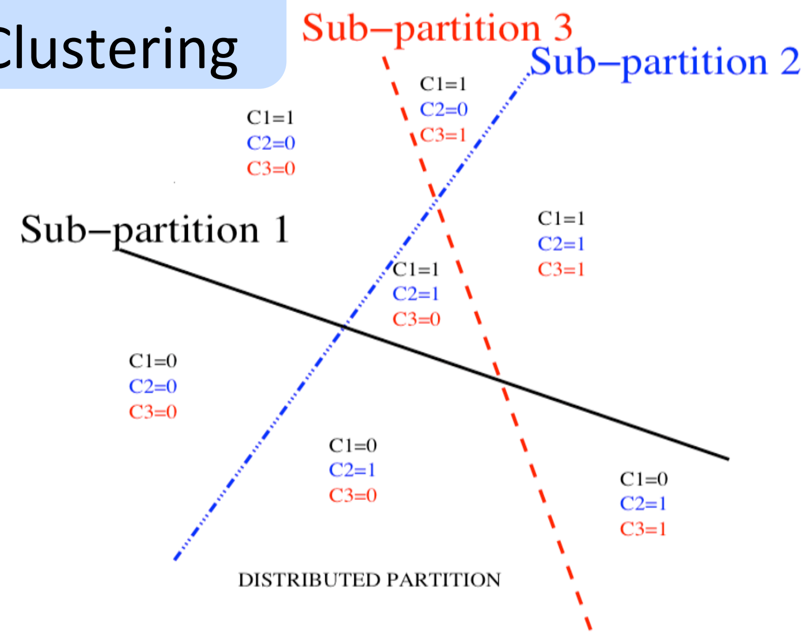



# The need for distributed representations

Clustering




Multi-Clustering



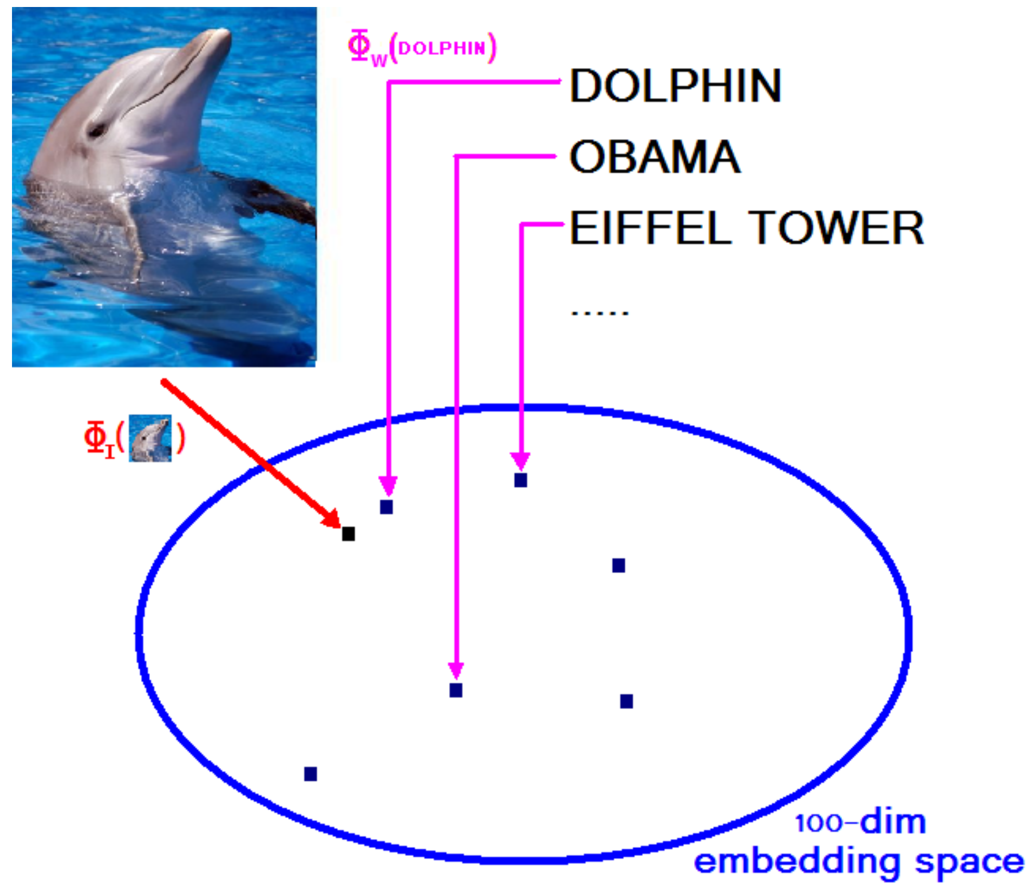

Learning a **set of features** that are not mutually exclusive can be exponentially more statistically efficient than nearest-neighbor-like or clustering-like models

# Google Image Search:
## Different object types represented in the 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.*

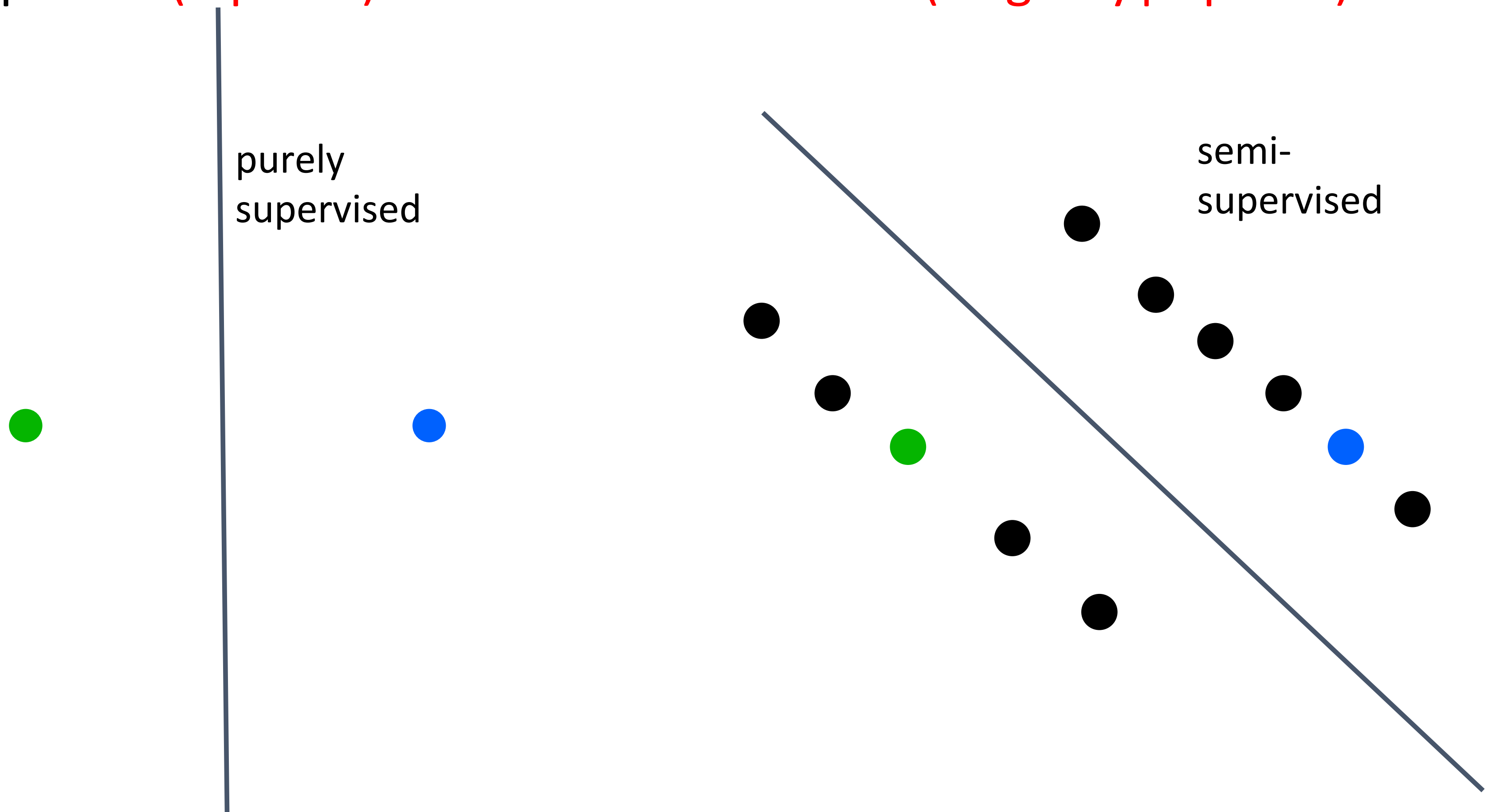
# How do humans generalize from very few examples?

- Brains may be born with ‘generic’ priors. Which ones?
- Humans **transfer** knowledge from previous learning:
  - Representations
  - Explanatory factors
- Previous learning from: unlabeled data

  + labels for other tasks

# Sharing Statistical Strength by Semi-Supervised Learning

prior: P(input=x) shares structure with P(target=y|input=x)



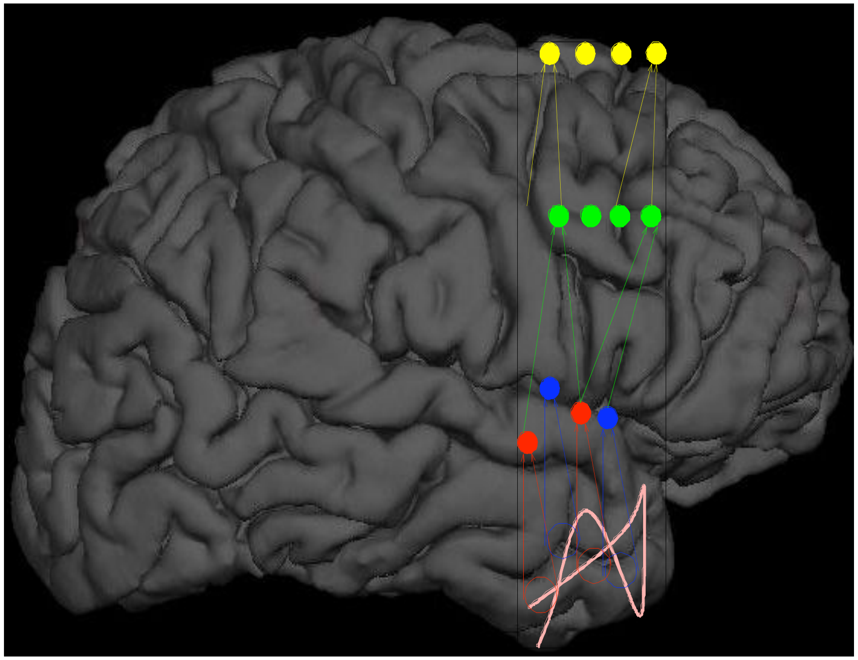
# Learning multiple levels of representation

Theoretical evidence for multiple levels of representation

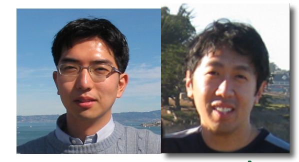
**Exponential gain for some families of functions**

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**

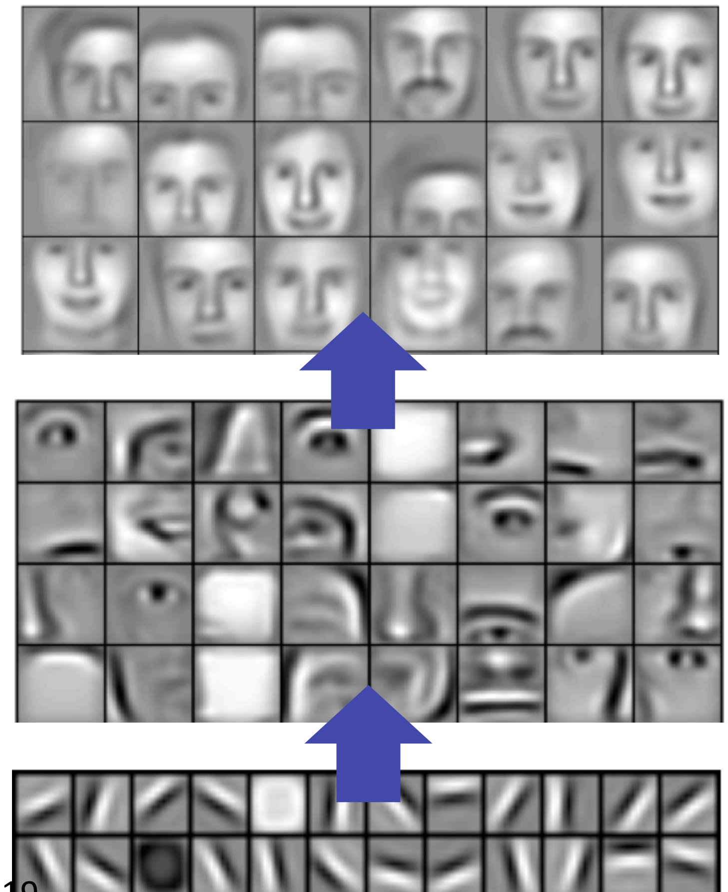


# Learning multiple levels of representation



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

Successive model layers learn deeper intermediate representations

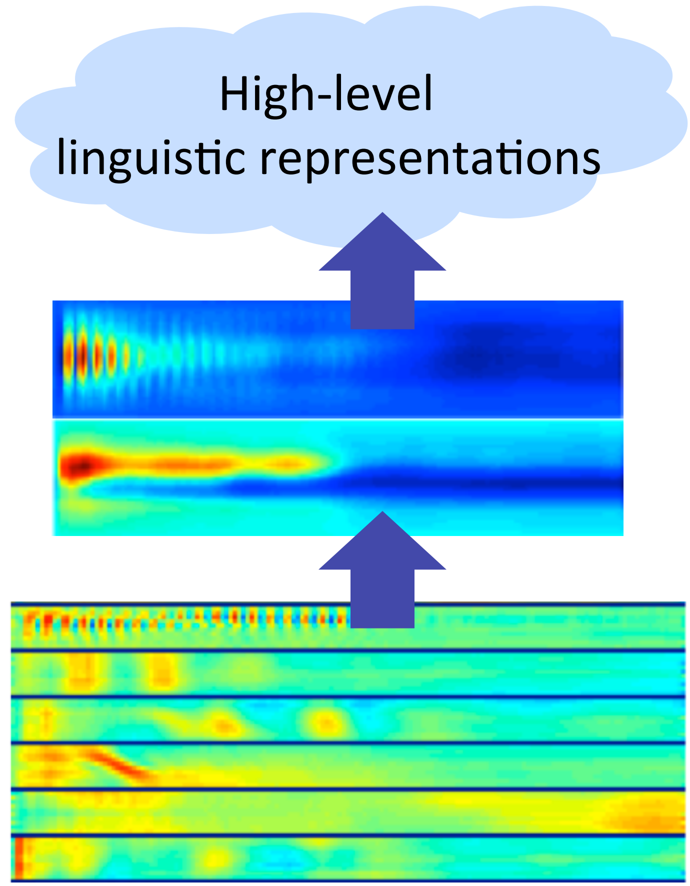


Layer 3

Parts combine to form objects

Layer 2

Layer 1





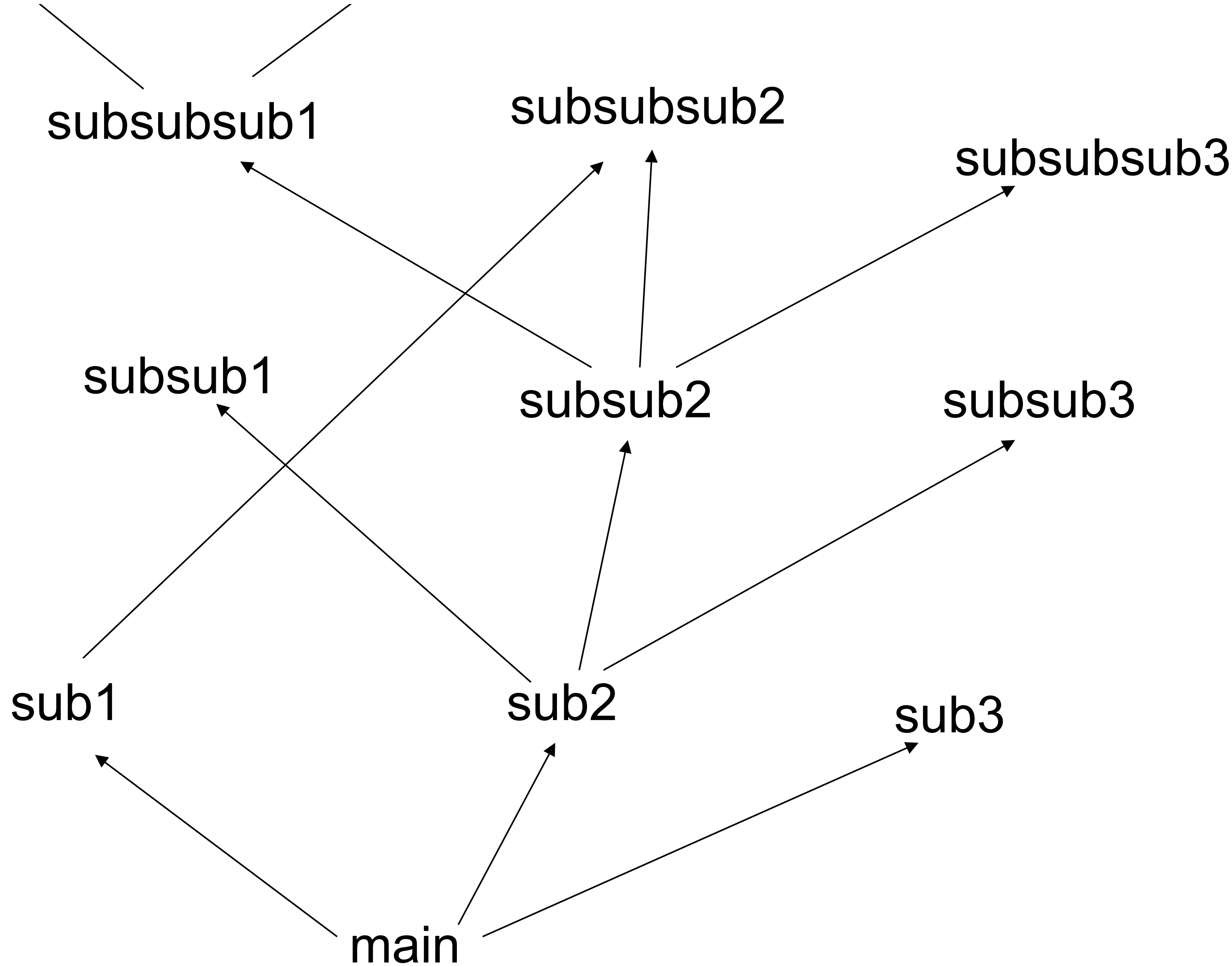

**Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction**

subroutine1 includes
subsub1 code and
subsub2 code and
subsubsub1 code

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

main

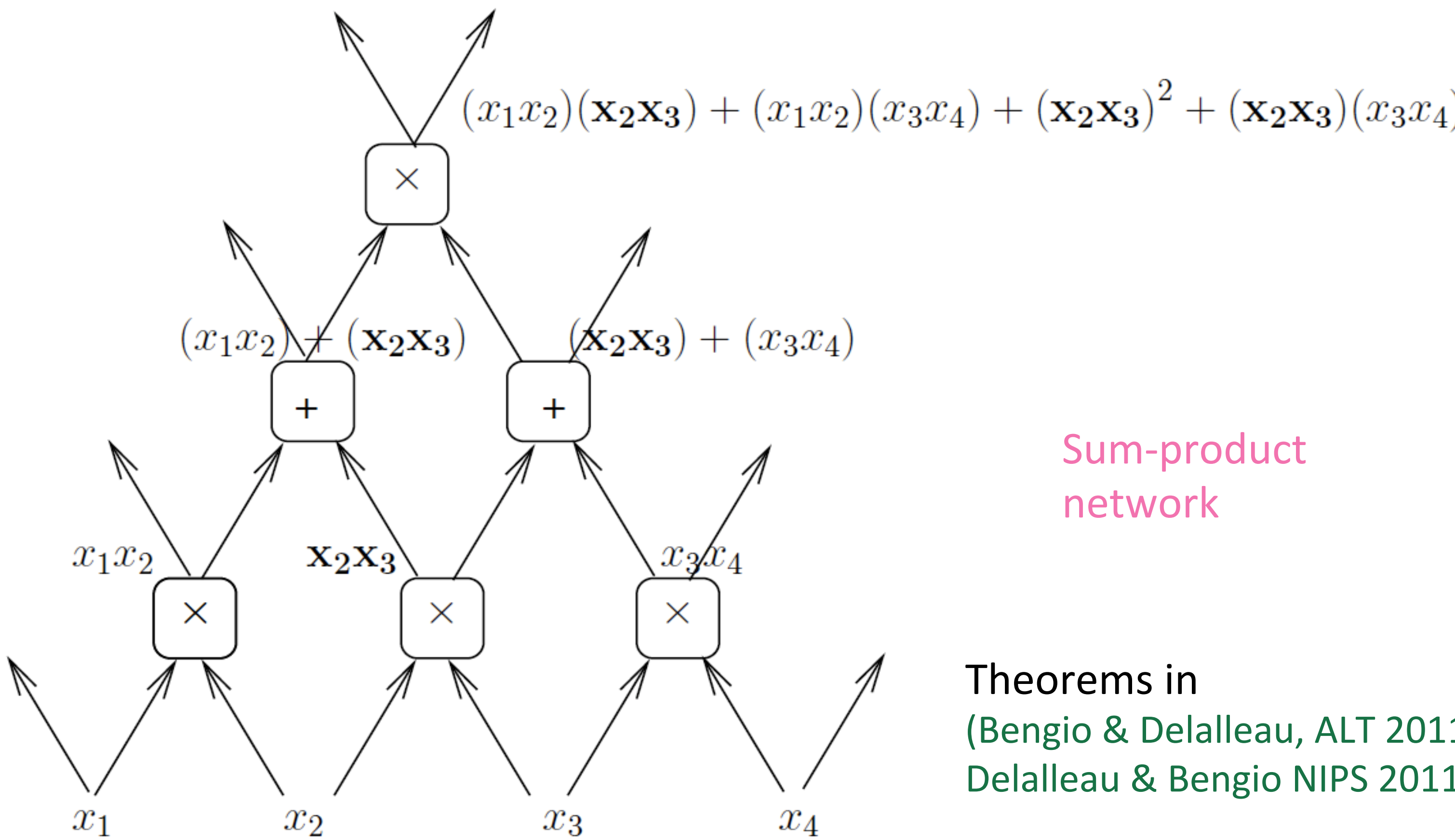
## “Shallow” computer program




## “Deep” computer program

# 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)

# Deep Networks for Speech Recognition: results from Google, IBM, MSR

| task | Hours of training data | Deep net+HMM | GMM+HMM same data | GMM+HMM more data |
|---|---|---|---|---|
| Switchboard | 309 | 16.1 | 23.6 | 17.1 (2k hours) |
| English Broadcast news | 50 | 17.5 | 18.8 | |
| Bing voice search | 24 | 30.4 | 36.2 | |
| Google voice input | 5870 | 12.3 | | 16.0 (lots more) |
| Youtube | 1400 | 47.6 | 52.3 | |



(numbers taken from Geoff Hinton's June 22, 2012 Google talk)

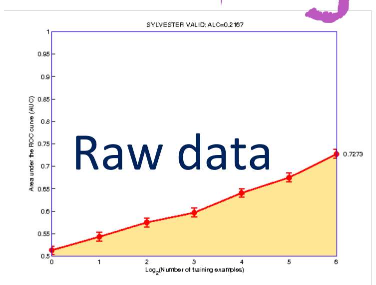
# Major Breakthrough in 2006

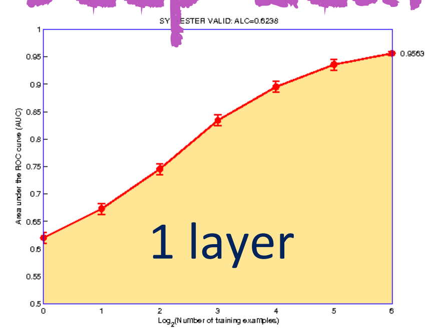
- Ability to train deep architectures by using layer-wise unsupervised learning, whereas previous purely supervised attempts had failed

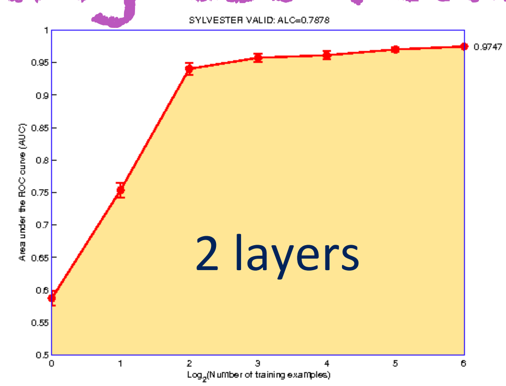
- Unsupervised feature learners:
  - RBMs
  - Auto-encoder variants
  - Sparse coding variants


Empirical successes since then: 2 competitions, Google, Microsoft, IBM...

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

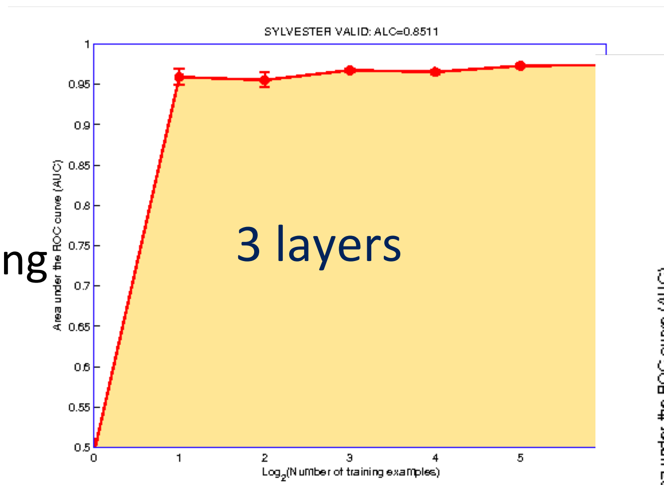


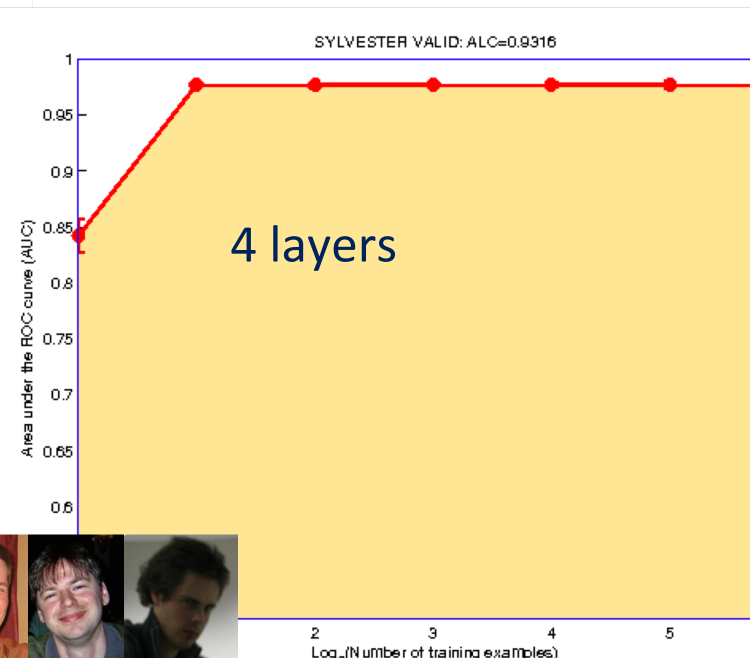




NIPS'2011 Transfer Learning Challenge Paper: ICML'2012

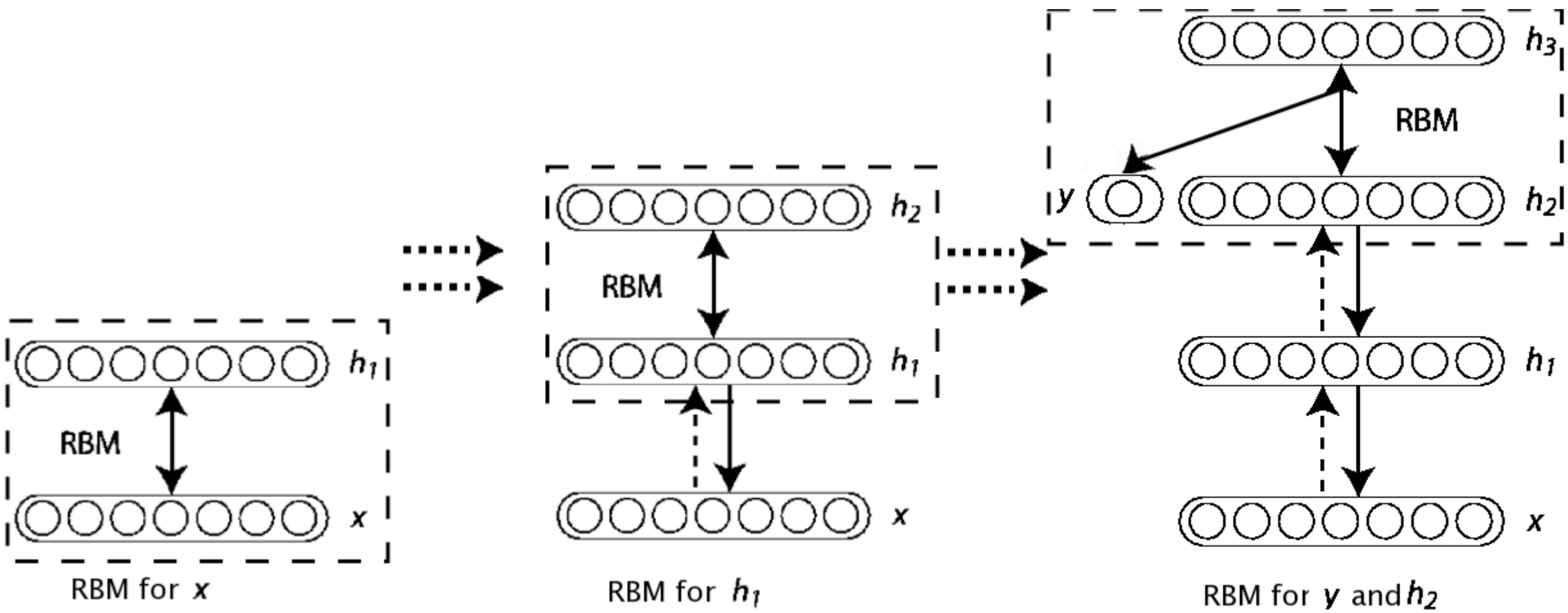
ICML'2011 workshop on Unsup. & Transfer Learning




# Stacking Single-Layer Learners

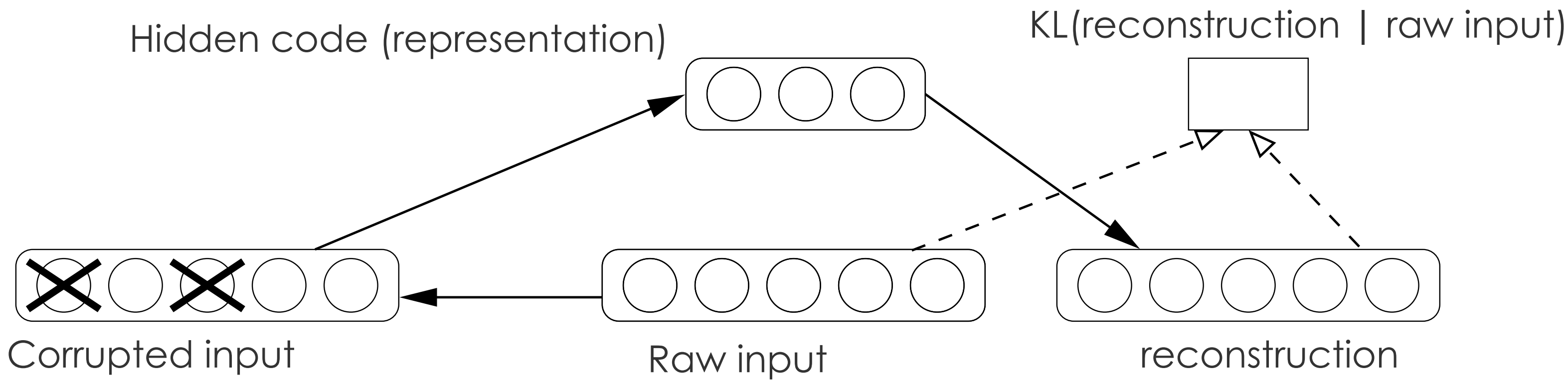
- One of the big ideas from Hinton et al. 2006: layer-wise unsupervised feature learning




Stacking Restricted Boltzmann Machines (RBM) → Deep Belief Network (DBN)

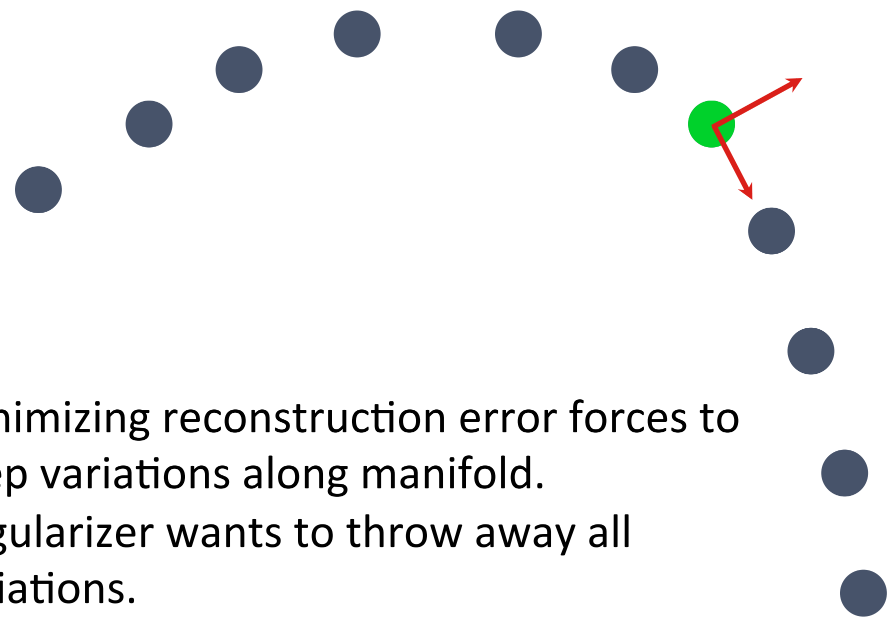
# Denoising Auto-Encoder
(Vincent et al 2008)

- Corrupt the input
- Try to reconstruct the uncorrupted input



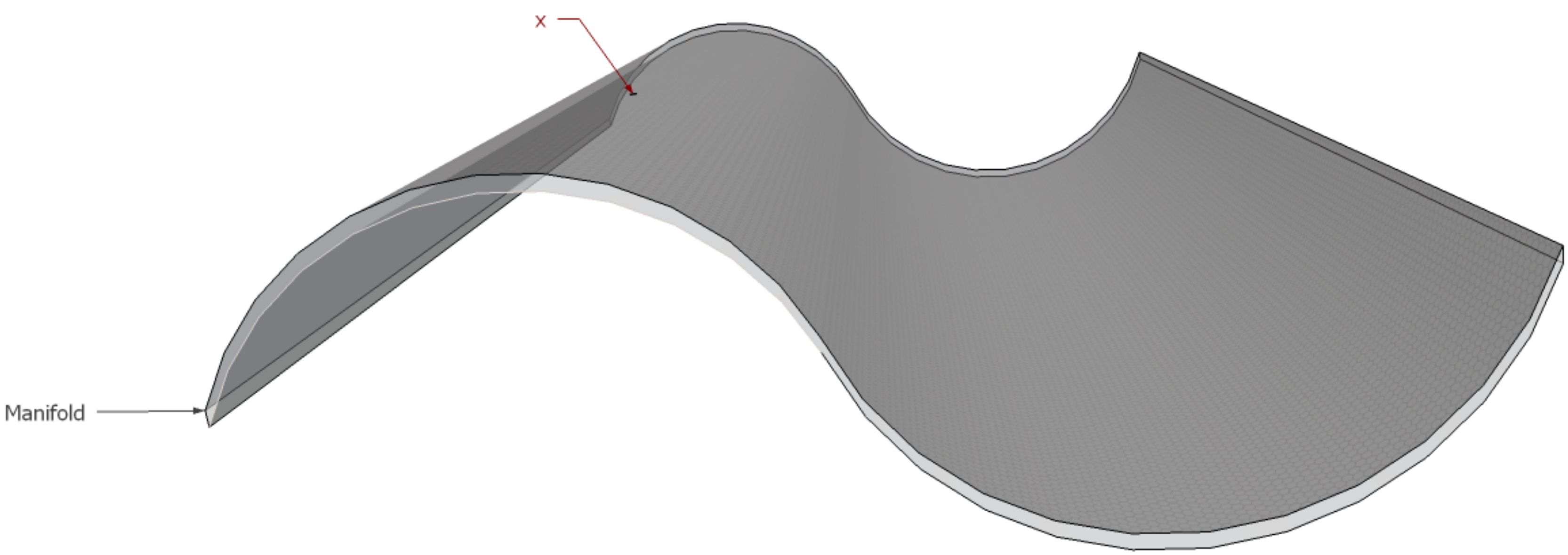

- Models the input density (through a form of score matching)

# Regularized Auto-Encoders Learn Salient Variations, like non-linear PCA with shared parameters

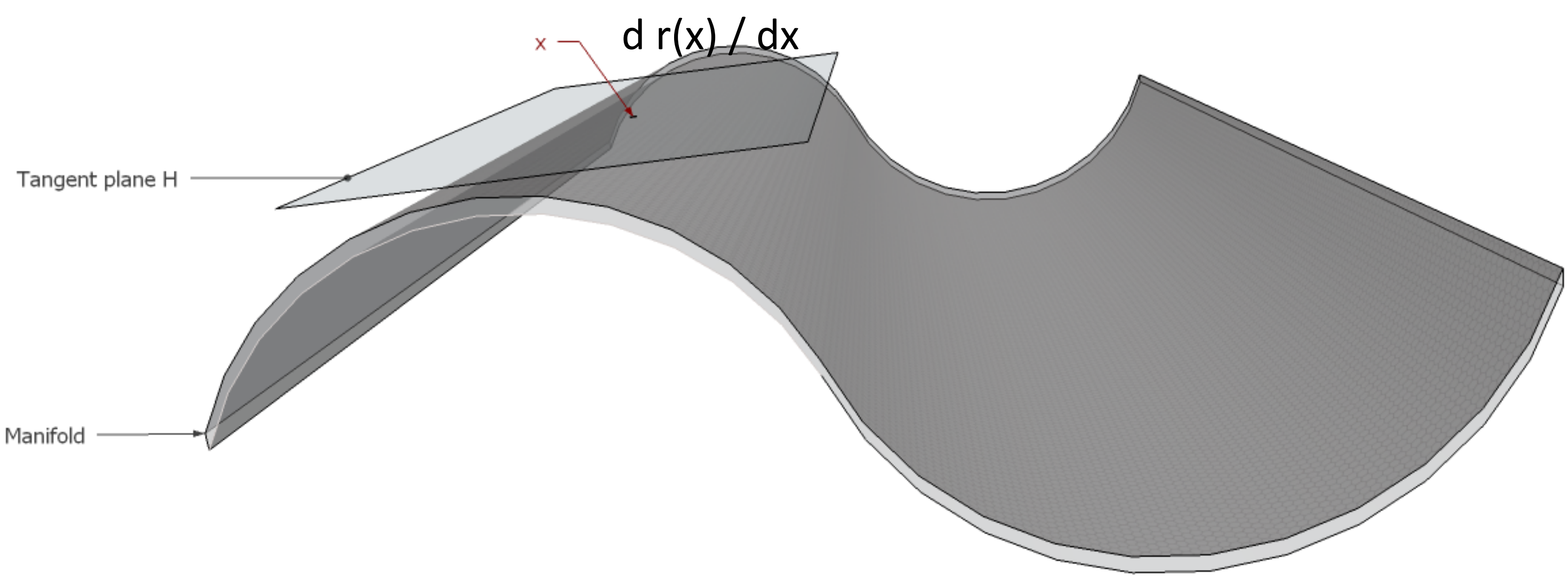


- Minimizing reconstruction error forces to keep variations along manifold.
- Regularizer wants to throw away all variations.
- With both: keep ONLY sensitivity to variations ON the manifold.

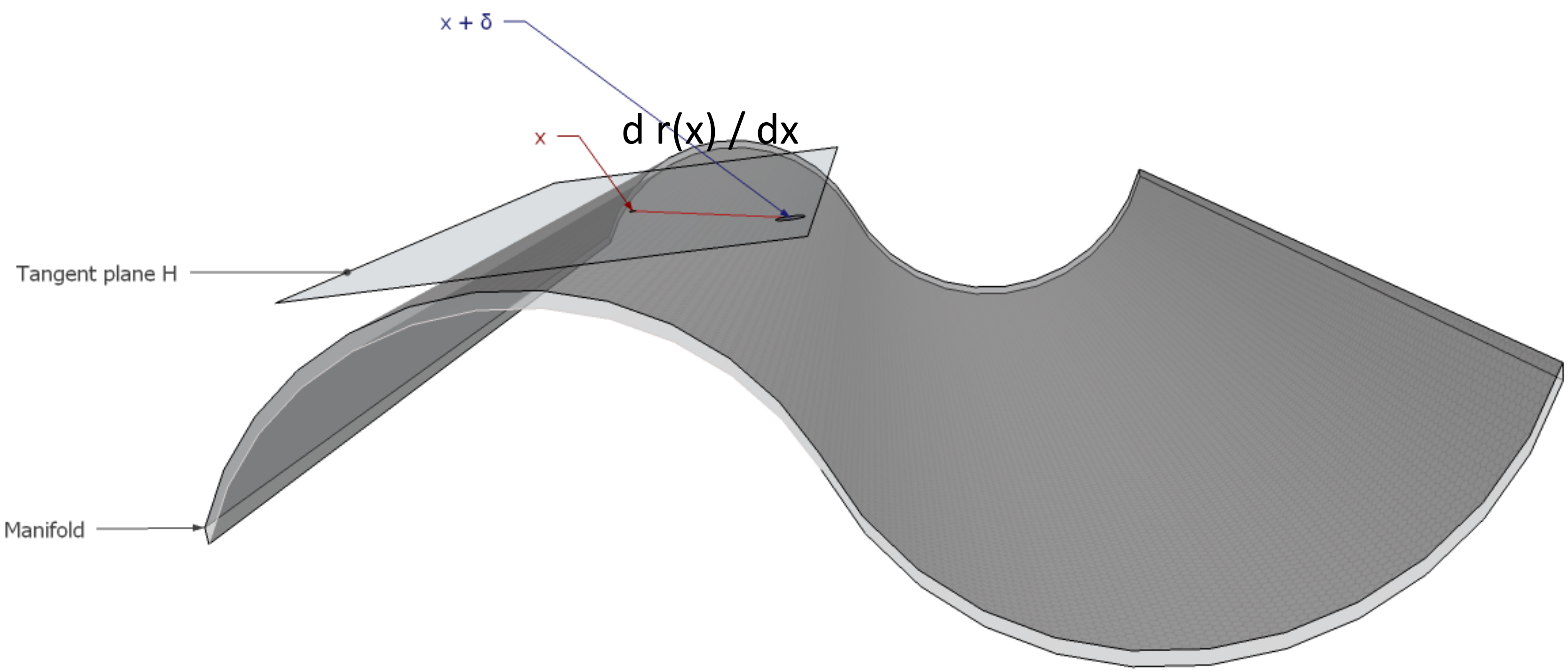
# Sampling from a Regularized Auto-Encoder (Rifai et al ICML 2012)



# Sampling from a Regularized Auto-Encoder (Rifai et al ICML 2012)

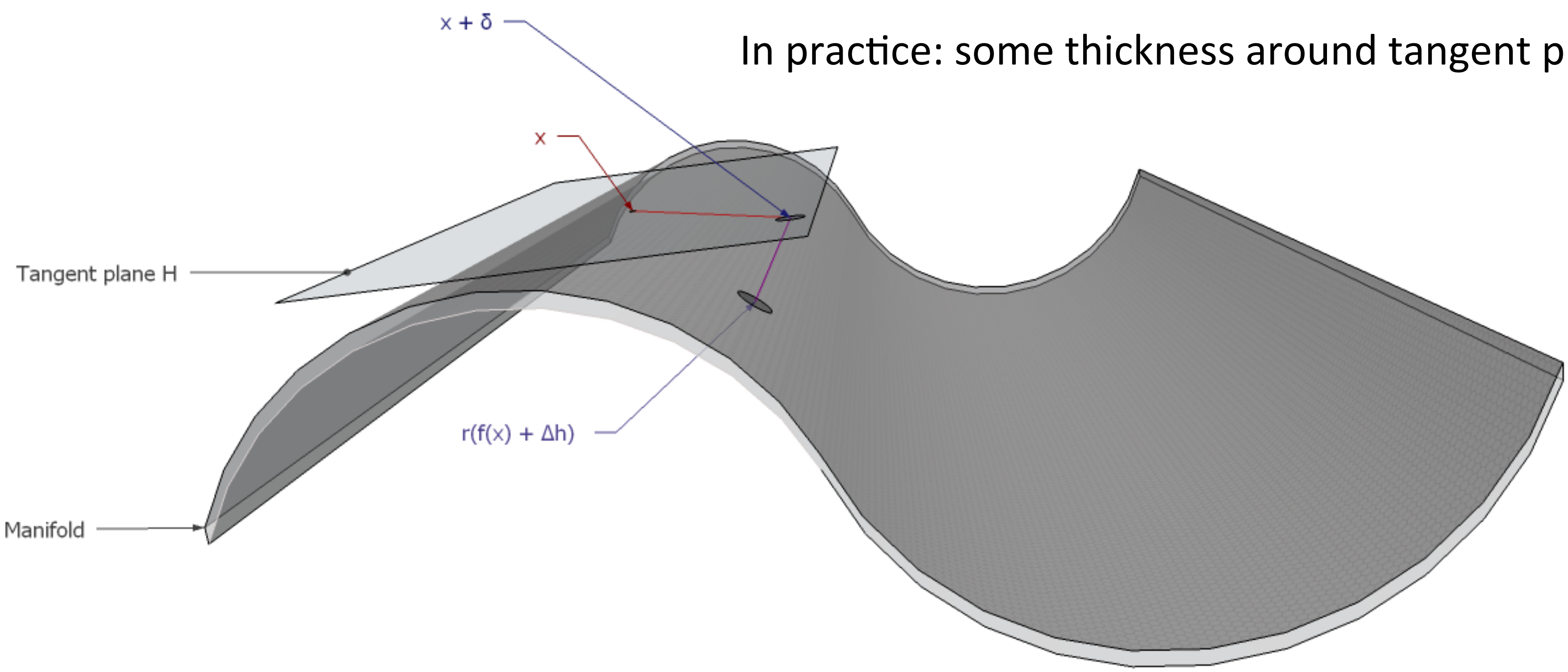


# Sampling from a Regularized Auto-Encoder
## (Rifai et al ICML 2012)



# Sampling from a Regularized Auto-Encoder
## (Rifai et al ICML 2012)

In practice: some thickness around tangent plane..



# Samples from a 2-level DAE

- TFD

- MNIST

# Invariance and Disentangling

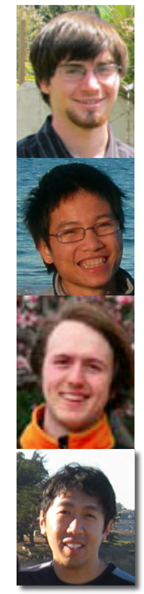
- Invariant features

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

  avoid the curse of dimensionality

# 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?**

# Sparse Representations

- Ask learned representation to be as sparse as possible
- Sparse → dense representations: entangles factors
- Easier to predict from
- Locally low-dimensional representation = local chart
- Hi-dim. sparse = efficient variable size representation

  = data structure

Few bits of information

Many bits of information

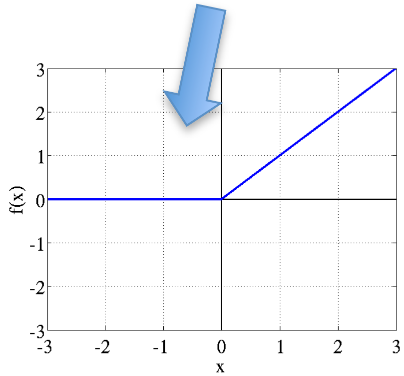
**Prior: only few concepts and attributes relevant per example**

# Deep Sparse Rectifier Neural Networks

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

**Neuroscience motivations**

Leaky integrate-and-fire model

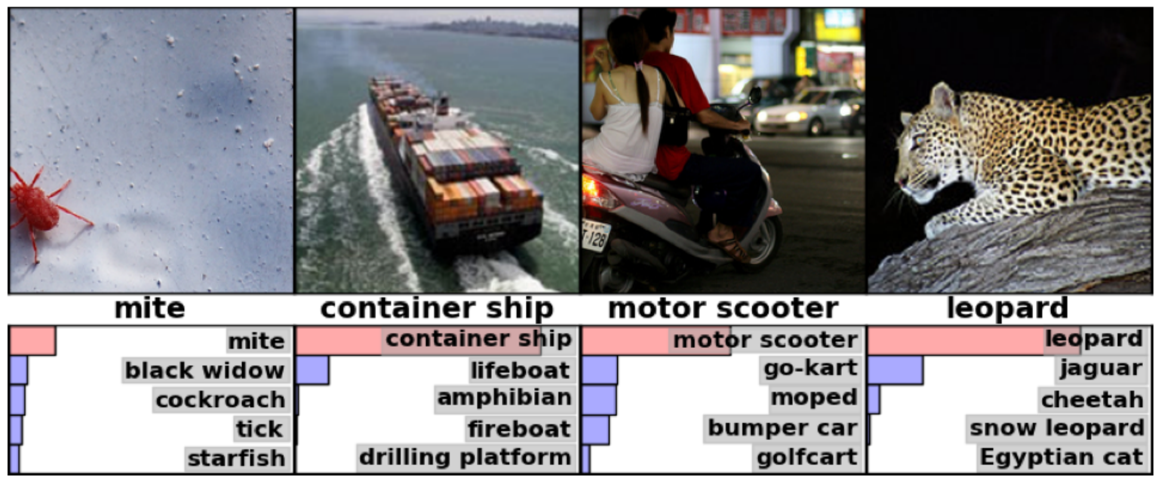


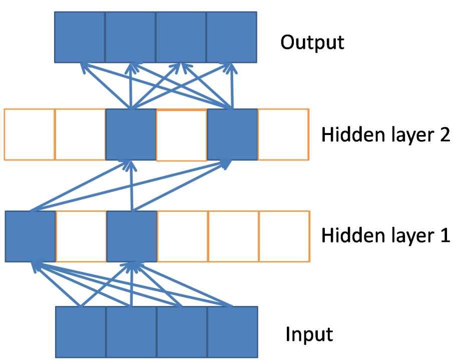

Rectifier

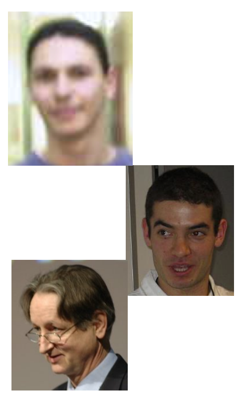
$f(x)=\max(0,x)$

**Machine learning motivations**

- ➡ Sparse representations
- ➡ Sparse gradients



**Outstanding results** by Krizhevsky et al 2012 killing the state-of-the-art on ImageNet 1000:

| | 1st choice | Top-5 |
|---|---|---|
| 2nd best | | 27% err |
| Previous SOTA | 45% err | 26% err |
| Krizhevsky et al | 37% err | 17% err |

# Stochastic Neurons as Regularizer: Improving neural networks by preventing co-adaptation of feature detectors (Hinton et al 2012, arXiv)

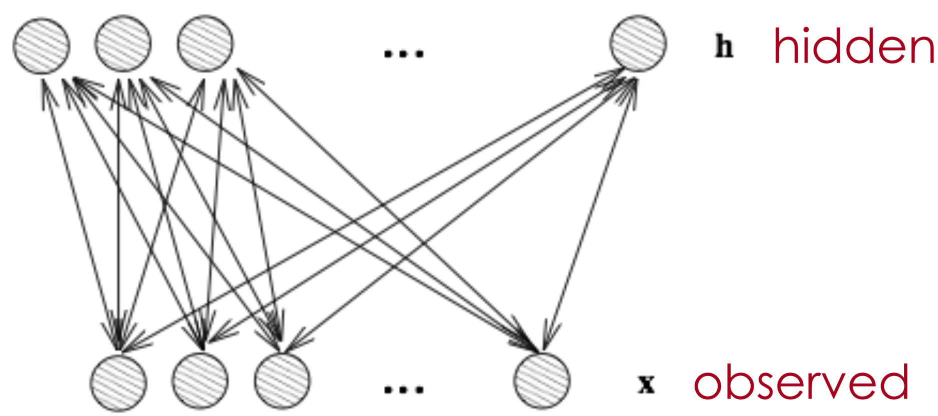
- **Dropouts** trick: during training multiply neuron output by random bit (p=0.5), during test by 0.5
- Similar to denoising auto-encoder, but corrupting every layer
- Equivalent to averaging over exponentially many architectures
  - Used by Krizhevsky et al to break through ImageNet SOTA
  - Also improves SOTA on CIFAR-10 (18→16% err)
  - Knowledge-free MNIST with DBMs (.95→.79% err)
  - TIMIT phoneme classification (22.7→19.7% err)

# Restricted Boltzmann Machine (RBM)

$$P(x,h) = \frac{1}{Z} e^{b^T h + c^T x + h^T W x} = \frac{1}{Z} e^{\sum_i b_i h_i + \sum_j c_j x_j + \sum_{i,j} h_i W_{ij} x_j}$$

A popular building block for deep architectures

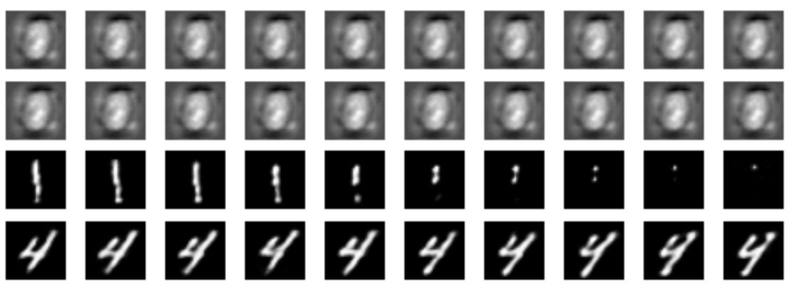
Needs to sample examples generated by the model during training



# Problems with Gibbs Sampling in RBMs

In practice, Gibbs sampling does not always mix well…

RBM trained by CD on MNIST



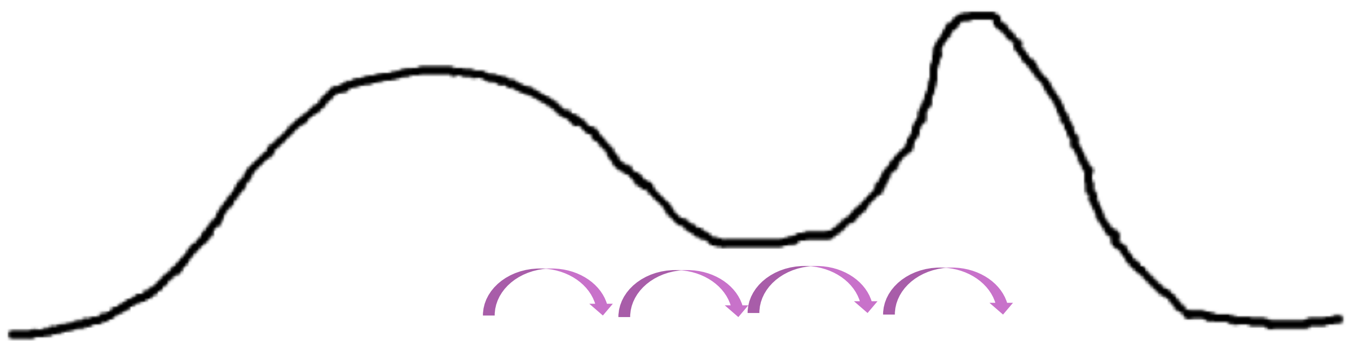
Chains from random state

Chains from real digits

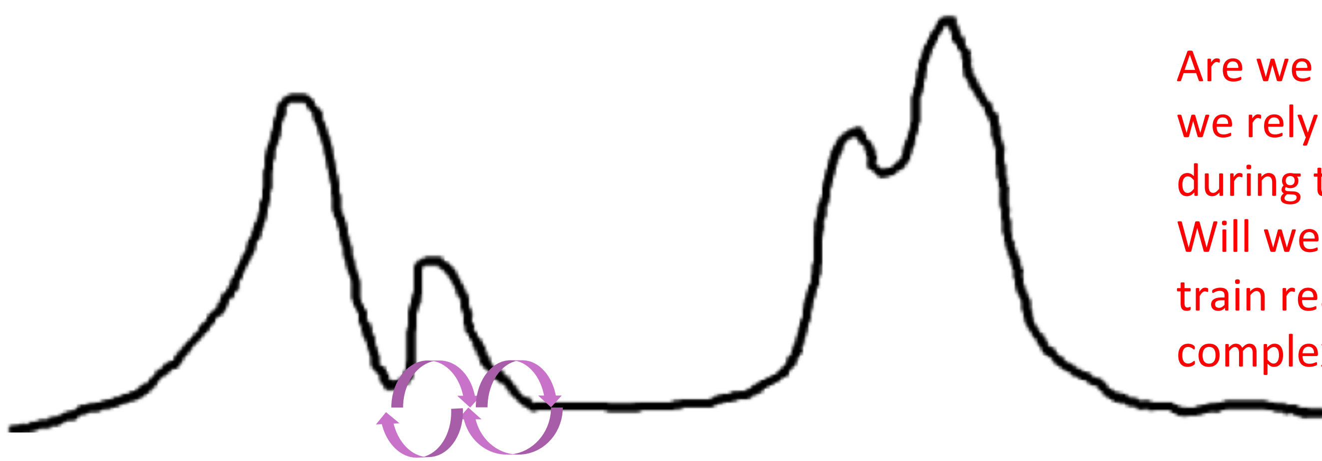
(Desjardins et al 2010)

# For gradient & inference: More difficult to mix with better trained models

- Early during training, density smeared out, mode bumps overlap



- Later on, hard to cross empty voids between modes



Are we doomed if we rely on MCMC during training? Will we be able to train really large & complex models?

# Poor Mixing: Depth to the Rescue

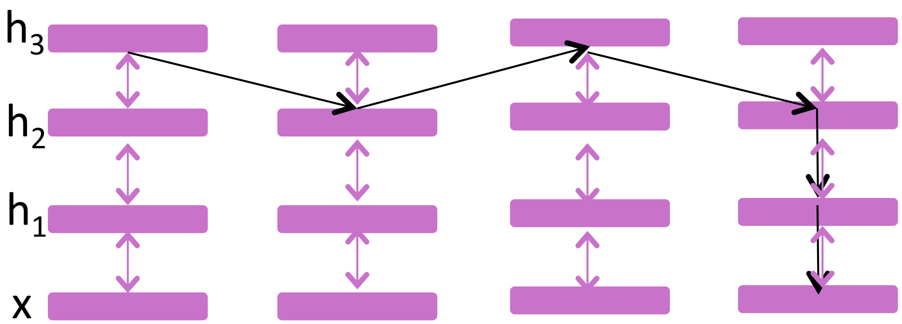
- Deeper representations can yield some disentangling
- Hypotheses:
  - more abstract/disentangled representations unfold manifolds and fill more the space
  - can be exploited for better mixing between modes
  - E.g. reverse video bit, class bits in learned object representations: easy to Gibbs sample between modes at abstract level

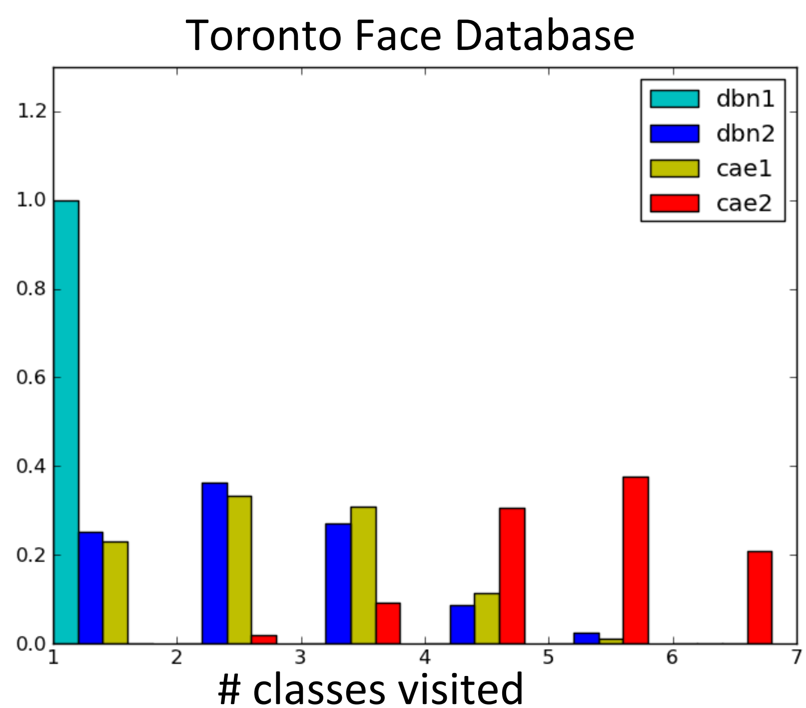
Layer
0
1
2

Points on the interpolating line between two classes, at different levels of representation

# Poor Mixing: Depth to the Rescue

- Sampling from DBNs and stacked Contrastive Auto-Encoders:
    1. MCMC sample from top-level singler-layer model
    2. Propagate top-level representations to input-level repr.
- Visits modes (classes) faster

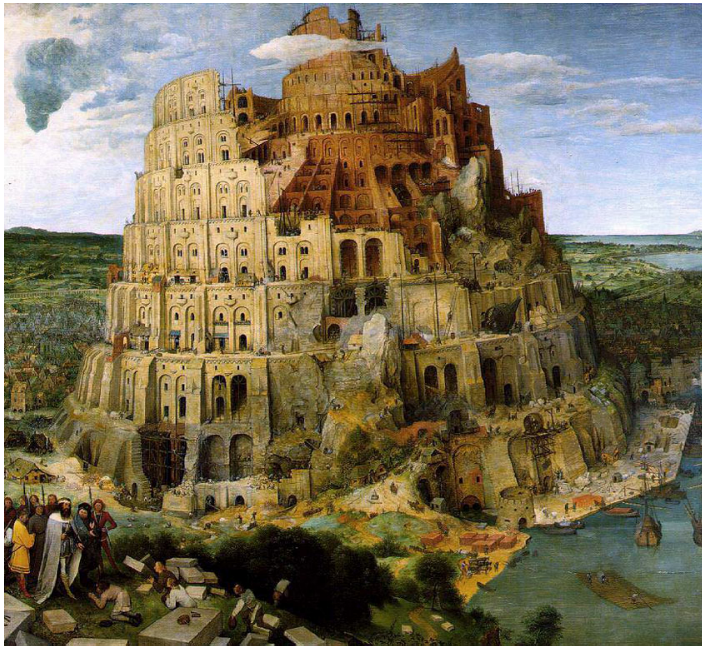


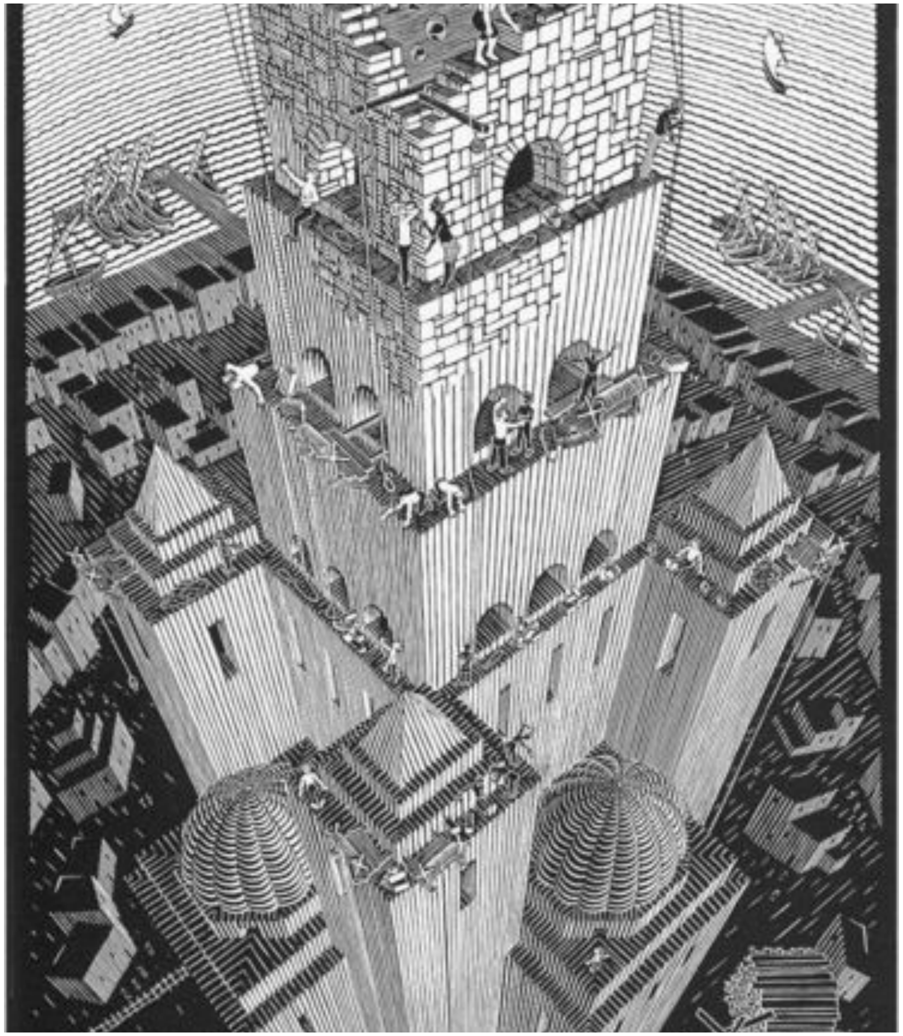
# Learning Multiple Levels of Abstraction

- The big payoff of deep learning is to allow learning higher levels of abstraction
- Higher-level abstractions disentangle the factors of variation, which allows much easier generalization and transfer
- More abstract representations

→Successful transfer (domains, languages), 2 international competitions won



# The End



LISA team: **Merci! Questions?**