Deep Learning: Theoretical Motivations

DLSS 2015

Deep Learning Summer School
Montreal, Canada

Yoshua Bengio

August 3, 2015

CIFAR
CANADIAN
INSTITUTE
FOR
ADVANCED
RESEARCH

Université
de Montréal

CENTRE
DE RECHERCHES
MATHÉMATIQUES

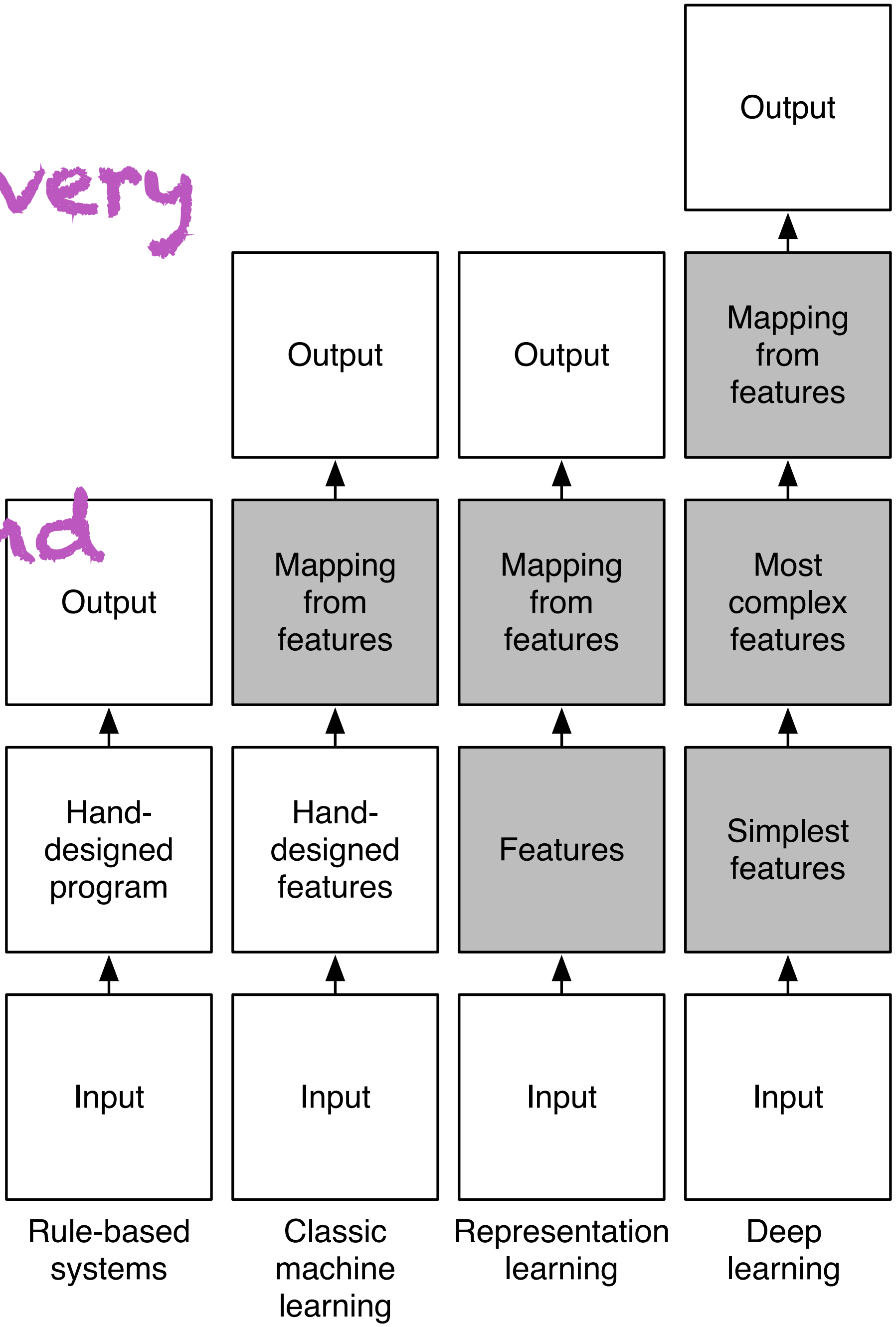
Breakthrough

- **Deep Learning**: **machine learning algorithms based on learning multiple levels of representation / abstraction.**

Amazing improvements in error rate in object recognition, object detection, speech recognition, and more recently, in natural language processing / understanding

Automating Feature Discovery

Discovering and representing higher-level abstractions



Why is deep learning working so well?

Machine Learning, AI & No Free Lunch

- Three key ingredients for ML towards AI

1. Lots & lots of data

2. Very flexible models

3. Powerful priors that can defeat the curse of dimensionality

Goal Hierarchy

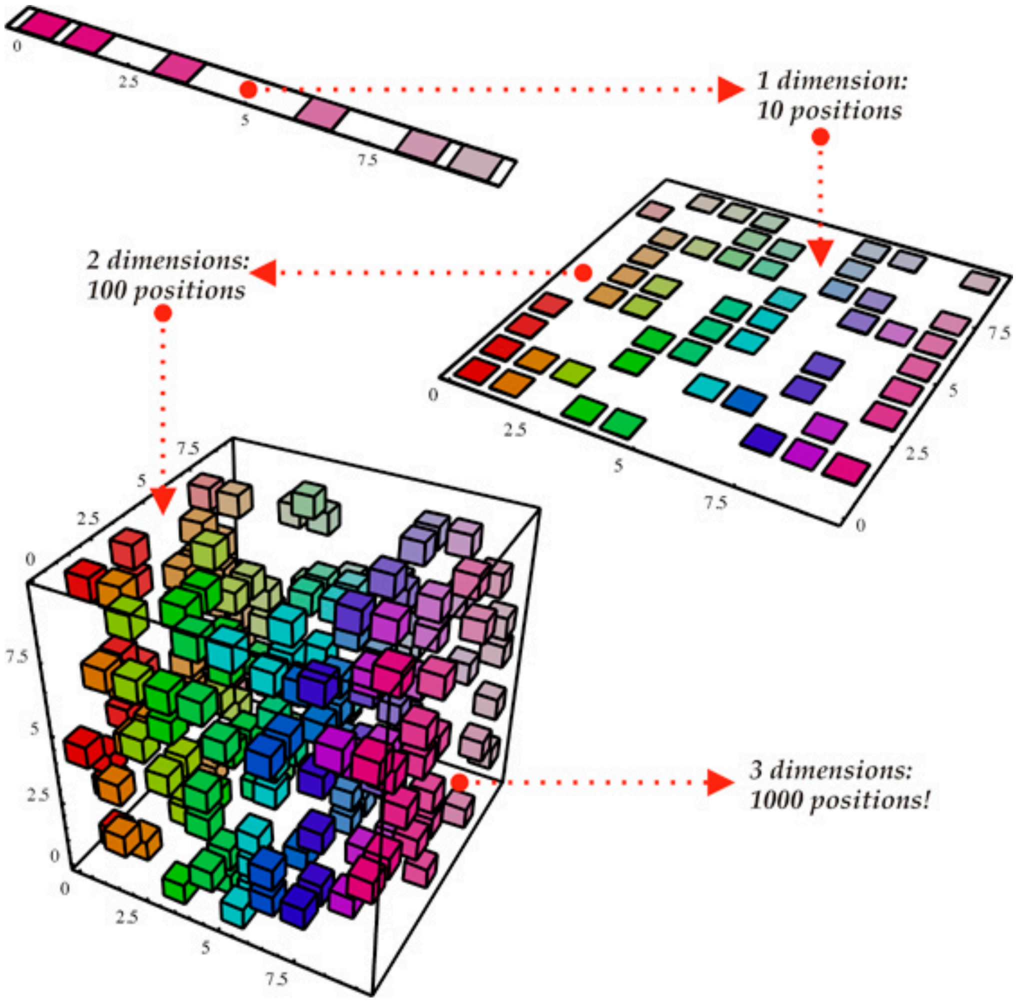
- **AI**
- Needs **knowledge**
- Needs **learning**
  (involves priors + ***optimization***/*search*)
- Needs **generalization**
  (guessing where probability mass concentrates)
- Needs ways to fight the curse of dimensionality
  (exponentially many configurations of the variables to consider)
- Needs disentangling the underlying explanatory factors
  (making sense of the data)

Why are classical non-parametric not cutting it?

ML 101. What We Are Fighting Against: The Curse of Dimensionality

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

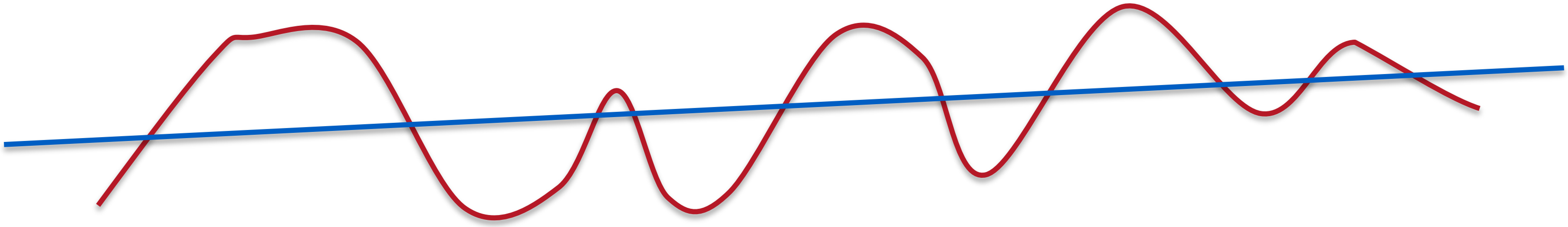
Classical solution: hope for a smooth enough target function, or make it smooth by handcrafting good features / kernel



Not Dimensionality so much as Number of Variations

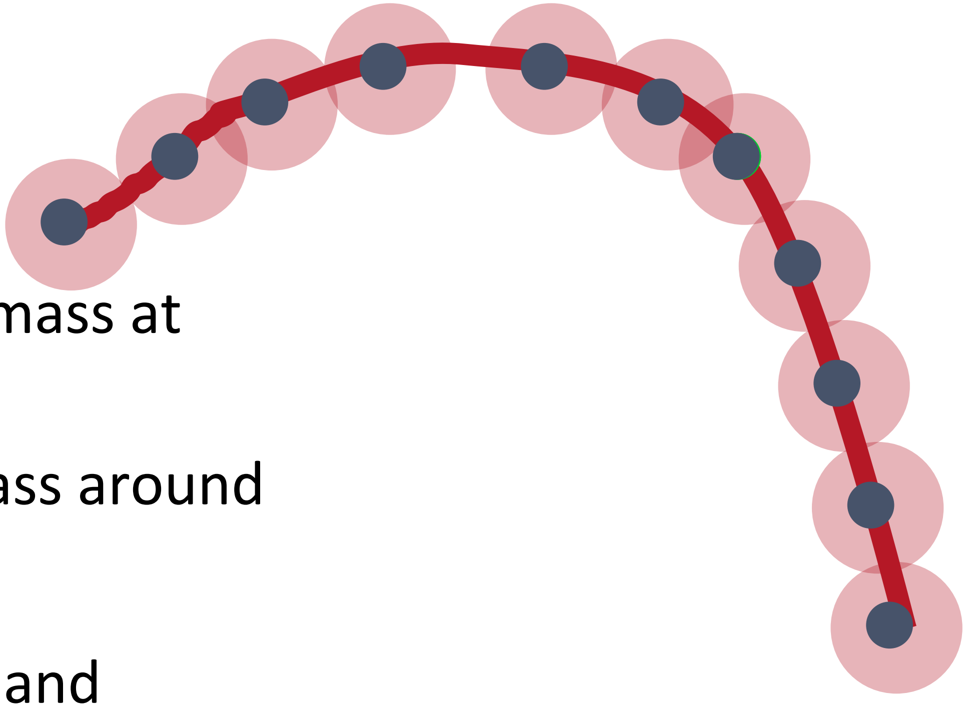
(Bengio, Dellalleau & Le Roux 2007)

- 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 some ‘structure’ and generalize accordingly

Bypassing the curse of dimensionality

We need to build compositionality into our ML models

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

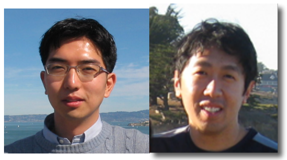
Exploiting compositionality gives an exponential gain in representational power

(1) Distributed representations / embeddings: feature learning

(2) Deep architecture: multiple levels of feature learning

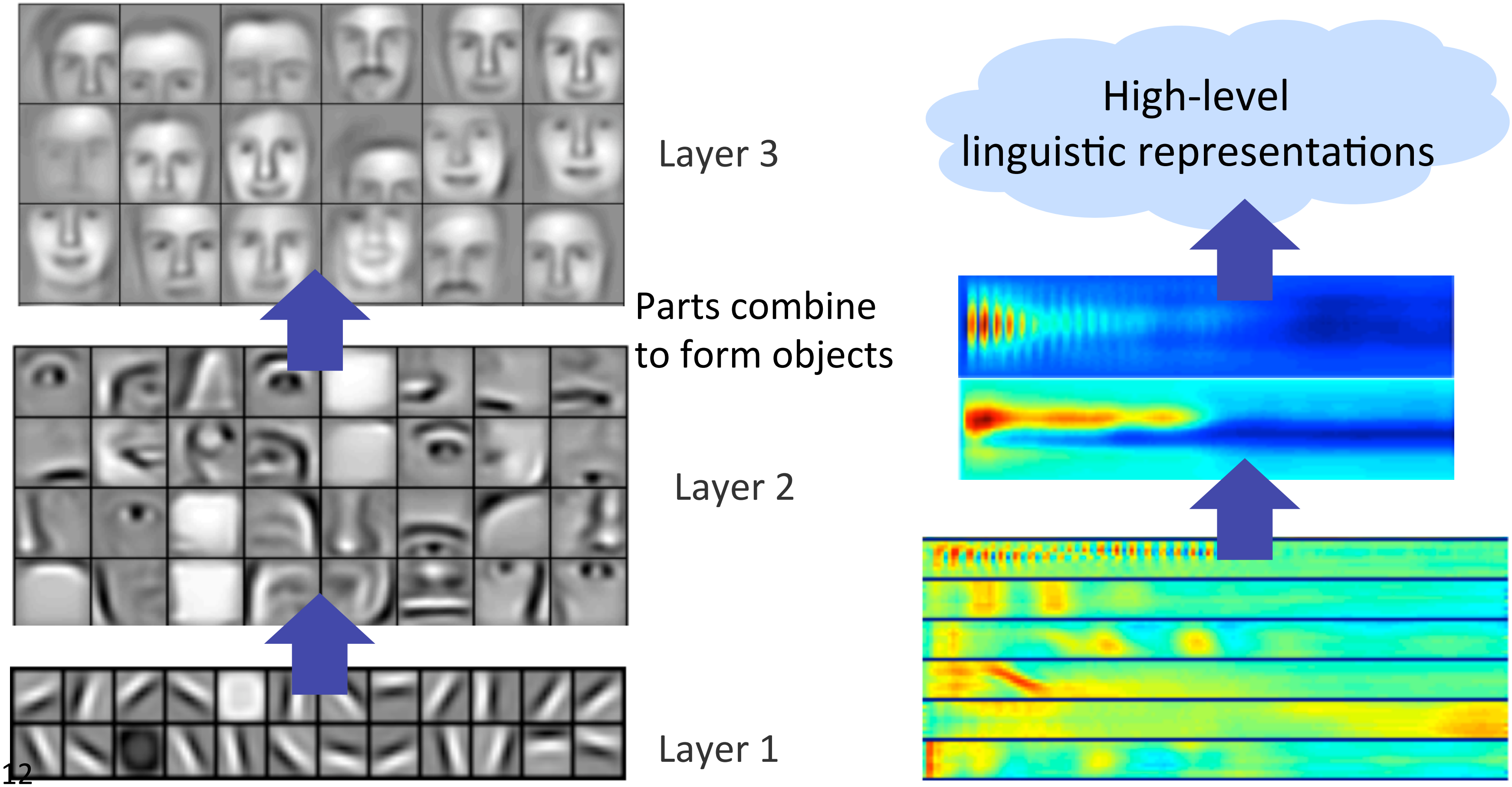
Additional prior: compositionality is useful to describe the world around us efficiently

Learning multiple levels of representation



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

Successive model layers learn deeper intermediate representations

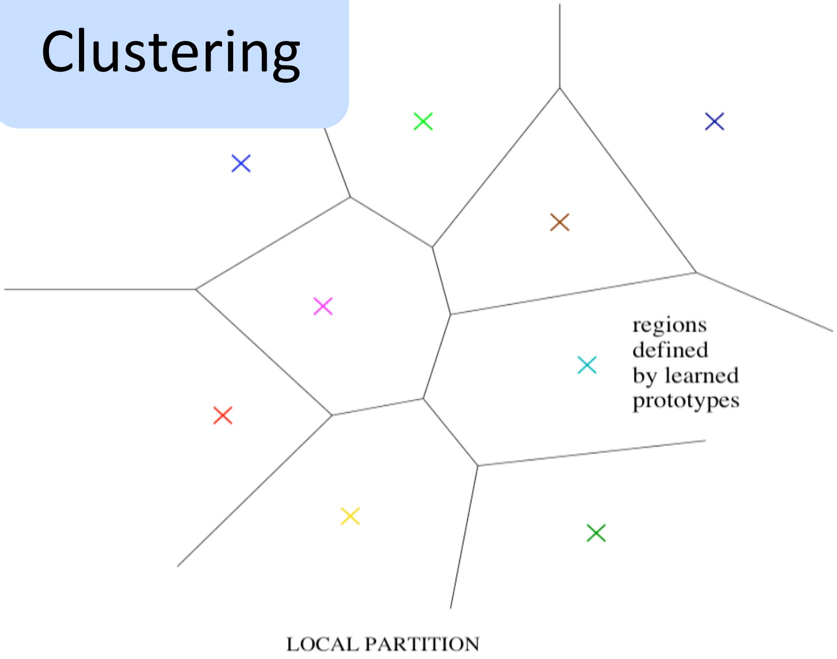


Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

The Power of Distributed Representations

Non-distributed representations

Clustering



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

- Parameters for each distinguishable region
- **# of distinguishable regions is linear in # of parameters**

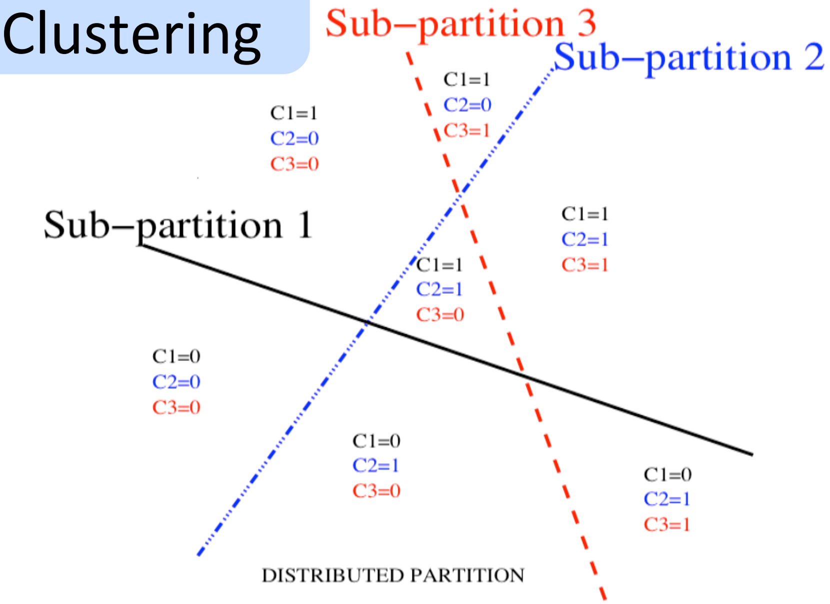
→ No non-trivial generalization to regions without examples

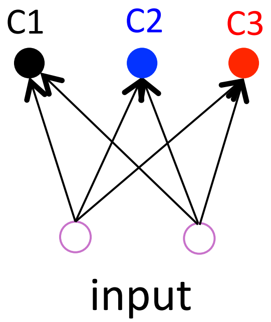
The need for distributed representations

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

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

Multi-Clustering





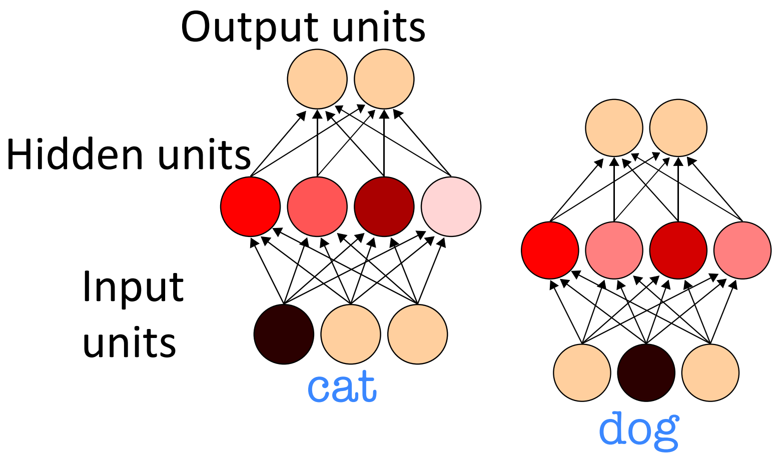
Non-mutually exclusive features/ attributes create a combinatorially large set of distinguiable configurations

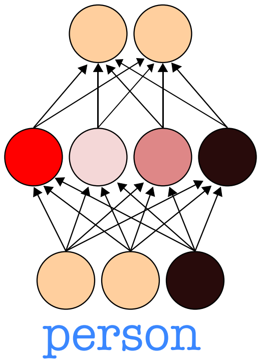
Classical Symbolic AI vs Representation Learning

- Two symbols are equally far from each other
- Concepts are not represented by symbols in our brain, but by patterns of activation

(Connectionism, 1980's)

Geoffrey Hinton

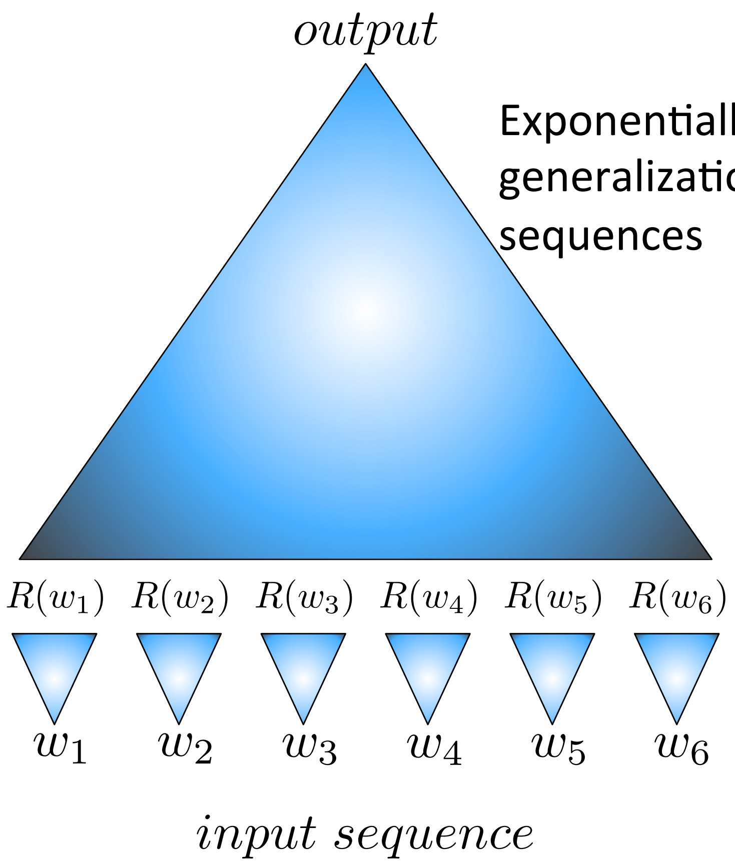




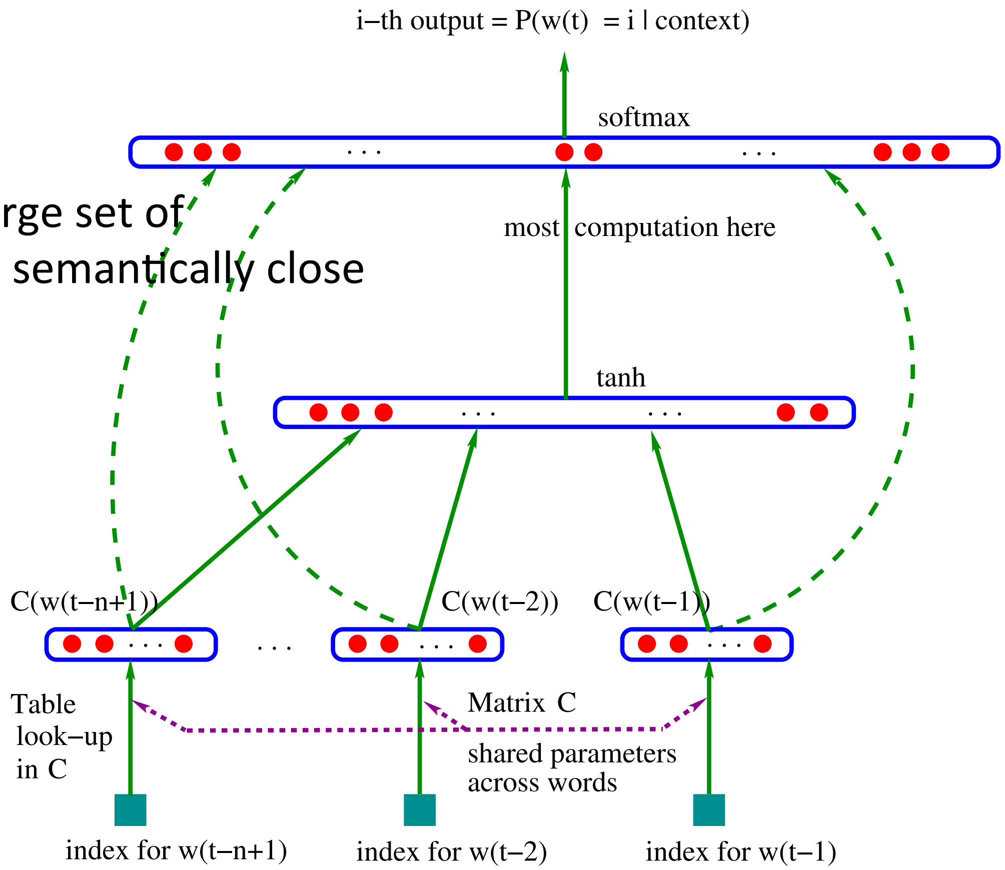
David Rumelhart

Neural Language Models: fighting one exponential by another one!

- (Bengio et al NIPS'2000)

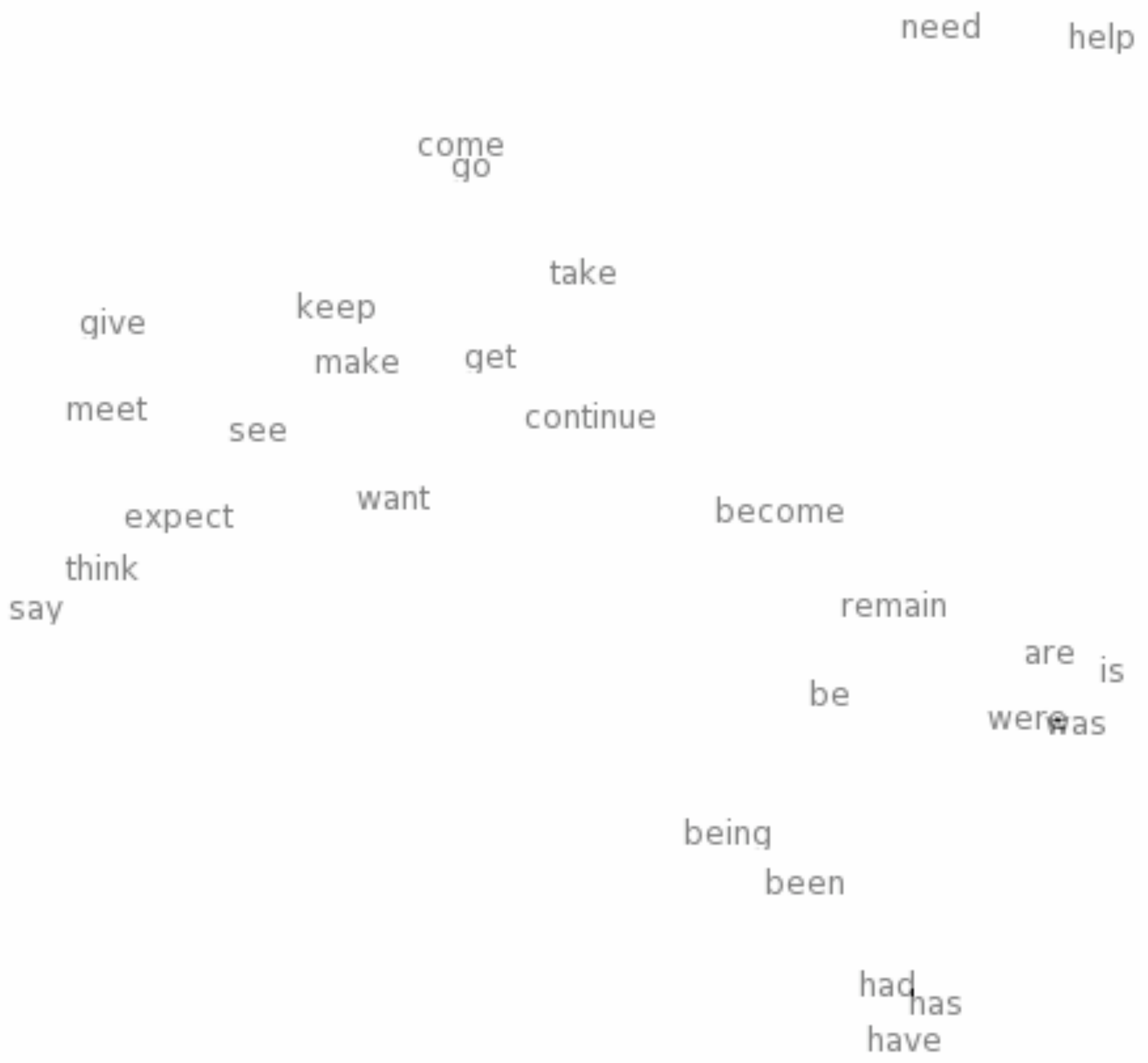


Exponentially large set of generalizations: semantically close sequences



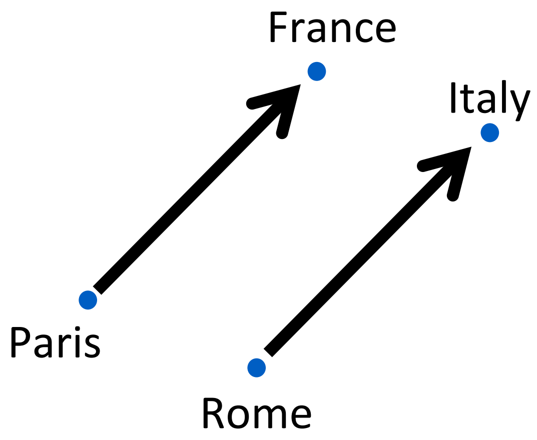
Exponentially large set of possible contexts

Neural word embeddings: visualization directions = Learned Attributes



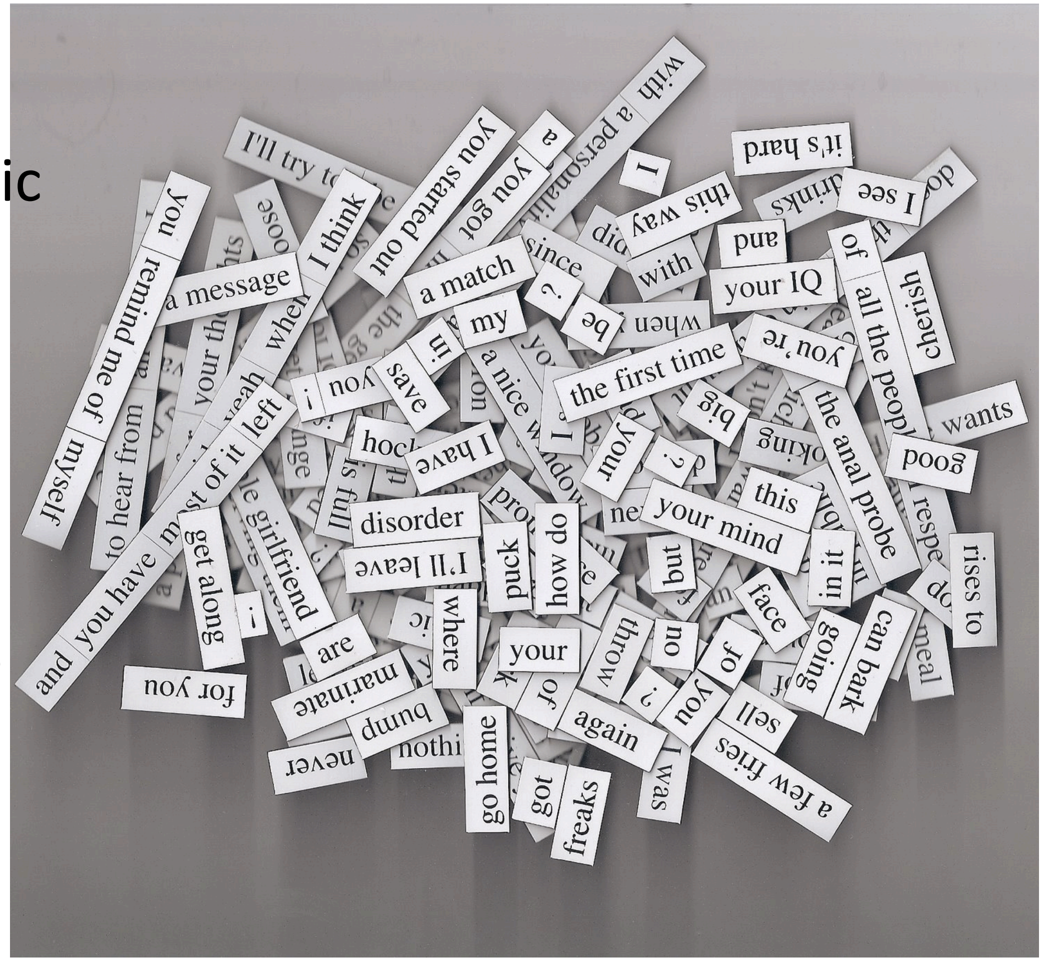
Analogical Representations for Free (Mikolov et al, ICLR 2013)

- Semantic relations appear as linear relationships in the space of learned representations
- King – Queen ≈ Man – Woman
- Paris – France + Italy ≈ Rome



The Next Challenge: Rich Semantic Representations for Word Sequences

- Impressive progress in capturing word semantics
  Easier learning: non-parametric (table look-up)
- Optimization challenge for mapping sequences to rich & complete representations
- Good test case: machine translation with auto-encoder framework



The Power of Deep Representations

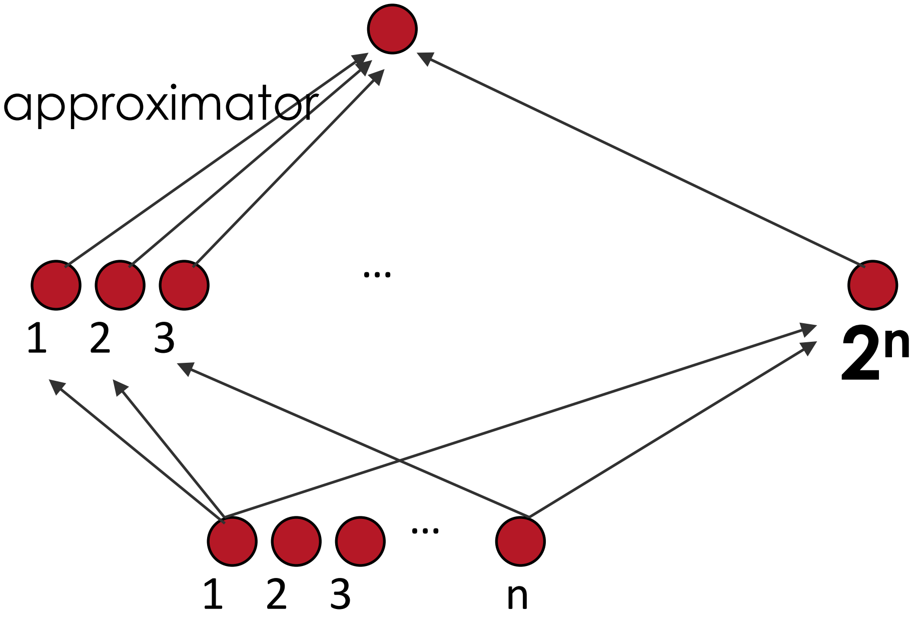
The Depth Prior can be Exponentially Advantageous

Theoretical arguments:

2 layers of
- Logic gates
- Formal neurons
- RBF units

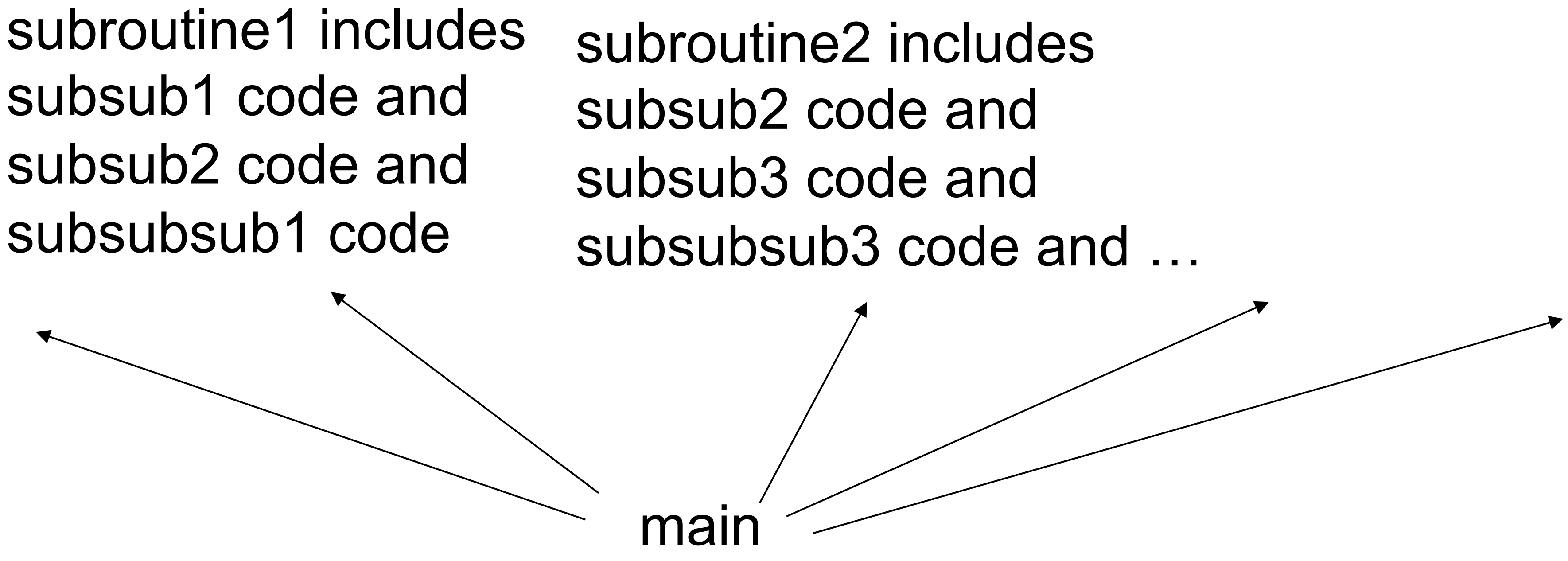
= universal approximator

RBMs & auto-encoders = universal approximator



Theorems on advantage of depth:
(Hastad et al 86 & 91, Bengio et al 2007, Bengio & Delalleau 2011, Braverman 2011, Pascanu et al 2014, Montufar et al **NIPS 2014**)

Some functions compactly represented with k layers may require exponential size with 2 layers

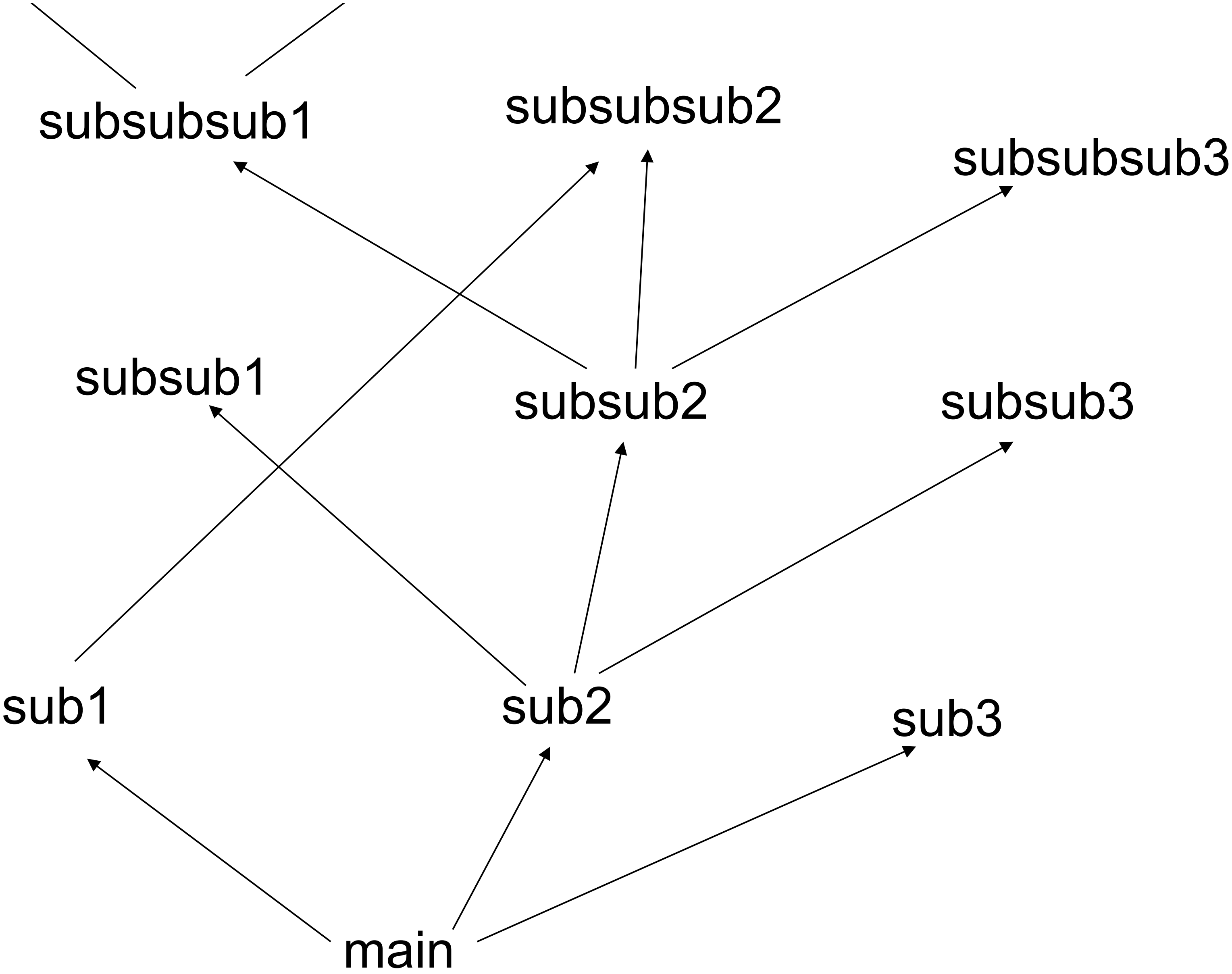


subroutine1 includes
subsub1 code and
subsub2 code and
subsubsub1 code

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

main

“Shallow” computer program



subsubsub1

subsubsub2

subsubsub3

subsub1

subsub2

subsub3

sub1

sub2

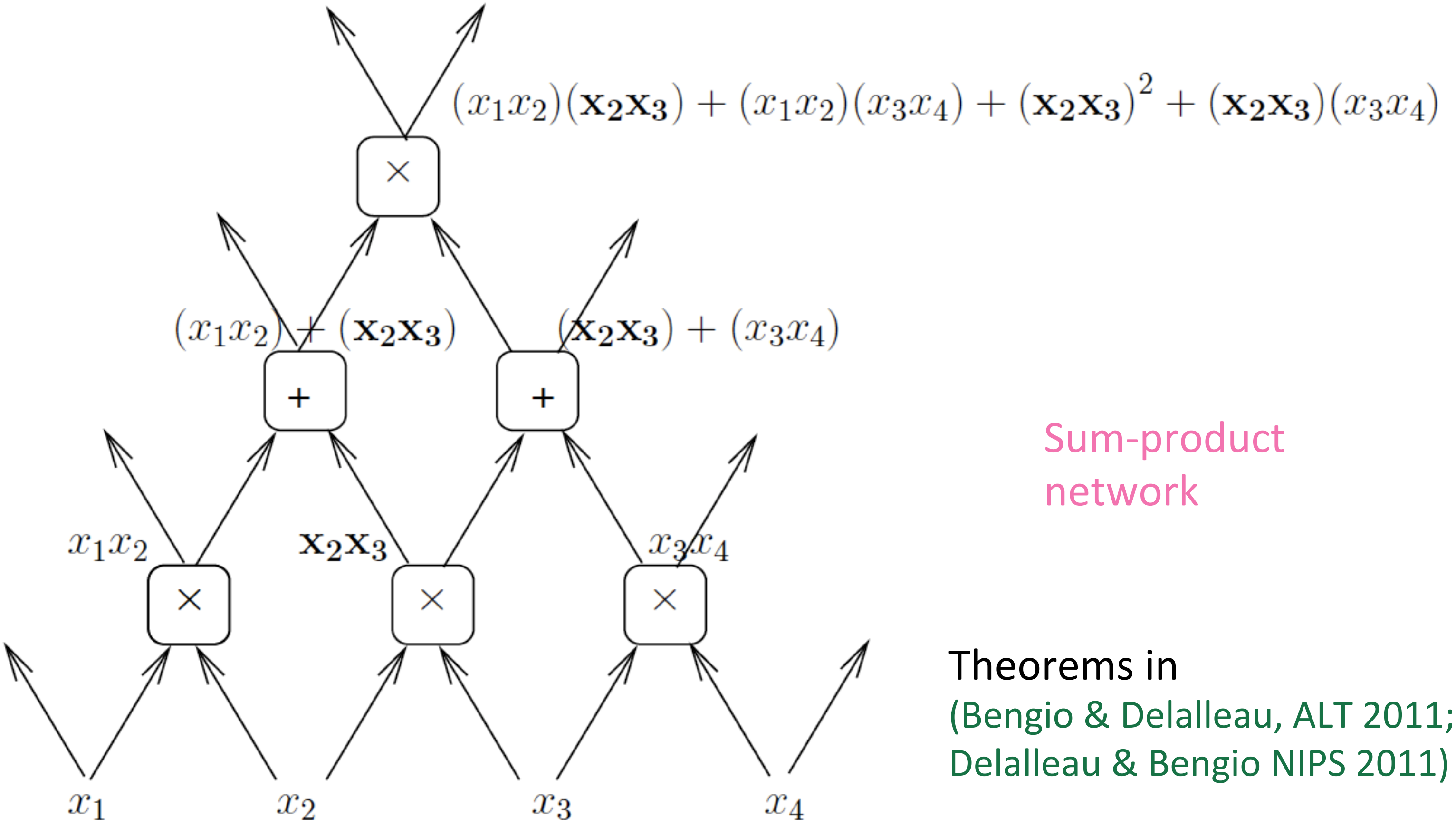
sub3

main

“Deep” computer program

Sharing Components in a Deep Architecture

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


$(x_1x_2)(\mathbf{x_2x_3}) + (x_1x_2)(x_3x_4) + (\mathbf{x_2x_3})^2 + (\mathbf{x_2x_3})(x_3x_4)$
×
$(x_1x_2) + (\mathbf{x_2x_3})$
$(\mathbf{x_2x_3}) + (x_3x_4)$
+
+
x_1x_2
$\mathbf{x_2x_3}$
x_3x_4
×
×
×
x_1
x_2
x_3
x_4

Sum-product network

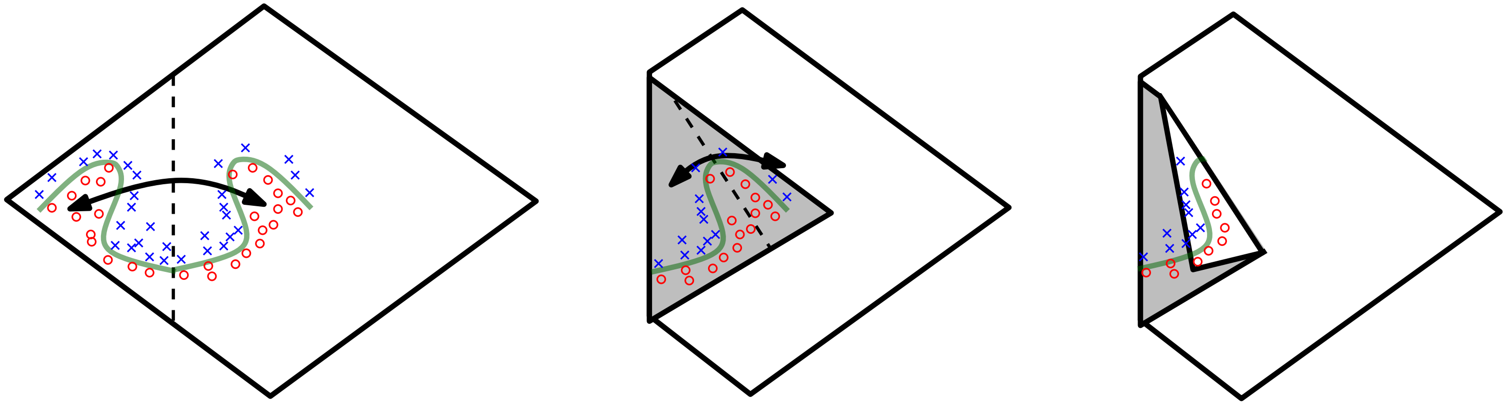
Theorems in (Bengio & Delalleau, ALT 2011; Delalleau & Bengio NIPS 2011)

New theoretical result: Expressiveness of deep nets with piecewise-linear activation fns

(Pascanu, Montufar, Cho & Bengio; ICLR 2014)

(Montufar, Pascanu, Cho & Bengio; NIPS 2014)

Deeper nets with rectifier/maxout units are exponentially more expressive than shallow ones (1 hidden layer) because they can split the input space in many more (not-independent) linear regions, with constraints, e.g., with abs units, each unit creates mirror responses, folding the input space:

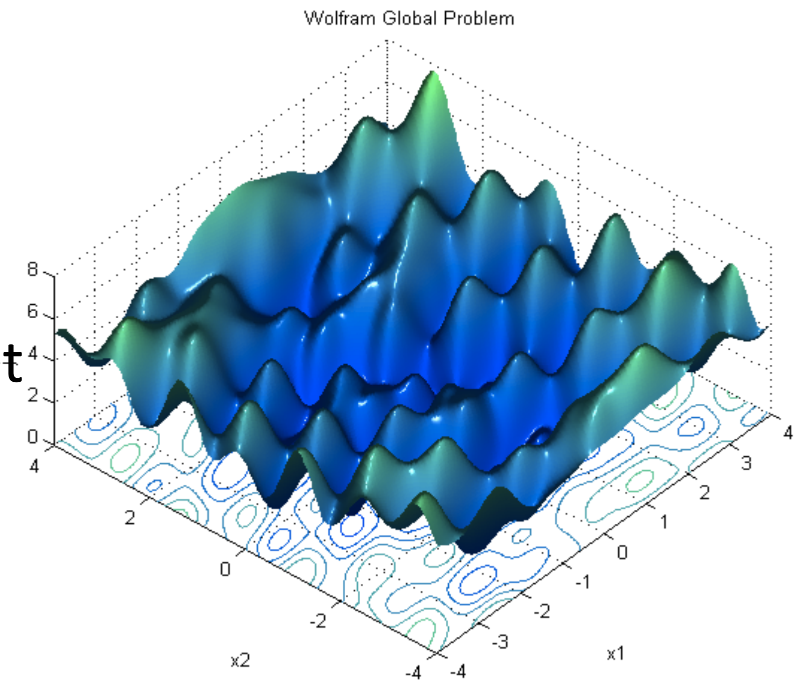


The Mirage of Convexity

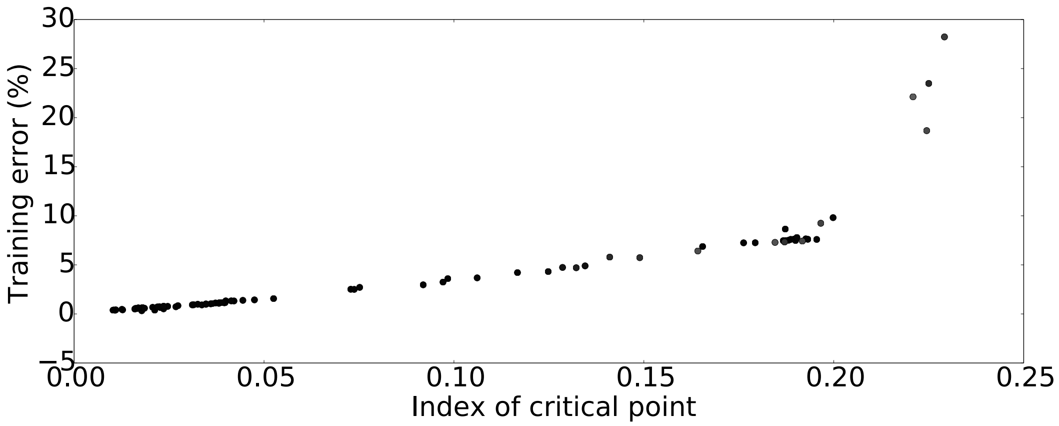
A Myth is Being Debunked: Local Minima in Neural Nets → Convexity is not needed

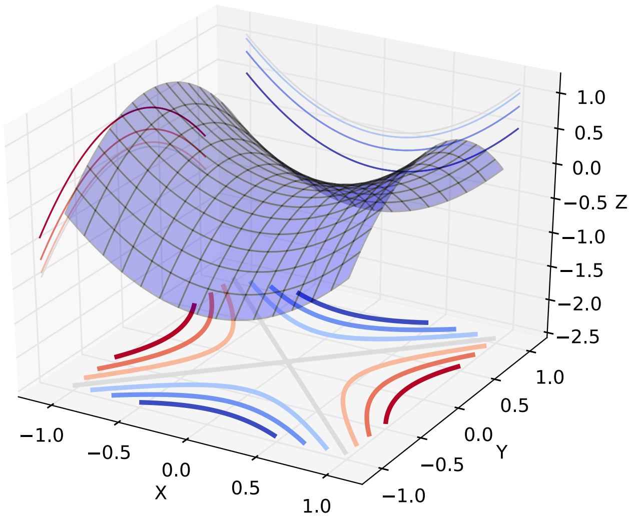
- (Pascanu, Dauphin, Ganguli, Bengio, arXiv May 2014): *On the saddle point problem for non-convex optimization*
- (Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio, NIPS' 2014): *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*
- (Choromanska, Henaff, Mathieu, Ben Arous & LeCun 2014): *The Loss Surface of Multilayer Nets*

Saddle Points



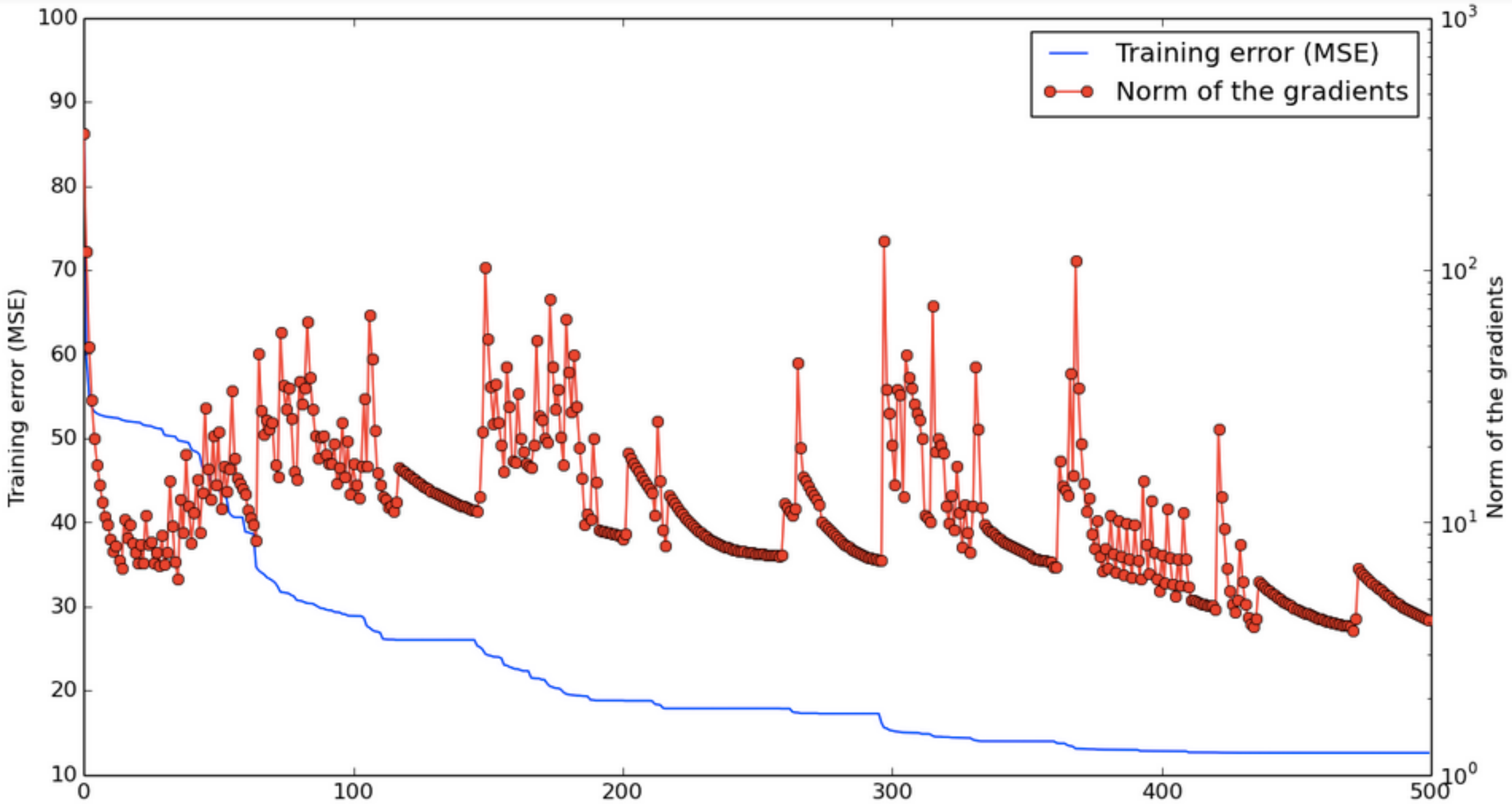
- Local minima dominate in low-D, but saddle points dominate in high-D
- Most local minima are close to the bottom (global minimum error)





Saddle Points During Training

- Oscillating between two behaviors:
  - Slowly approaching a saddle point
  - Escaping it

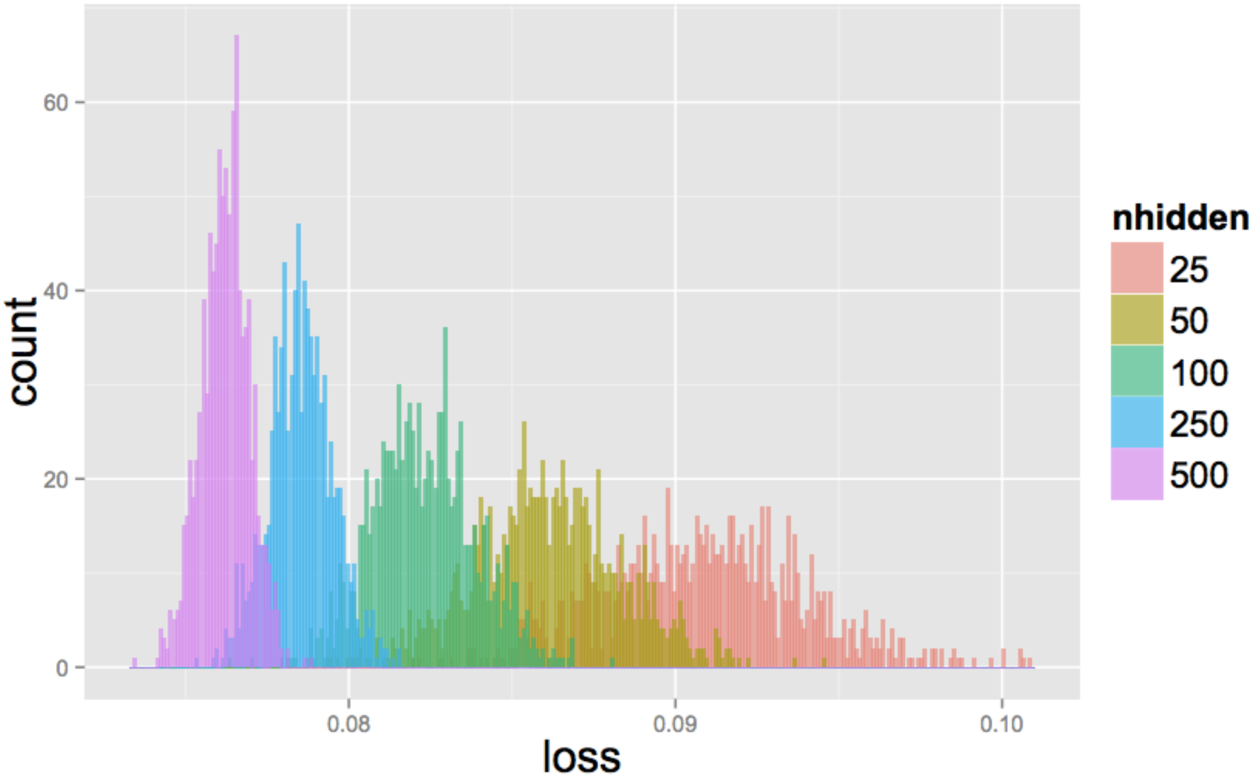


Low Index Critical Points

Choromanska et al & LeCun 2014, 'The Loss Surface of Multilayer Nets'

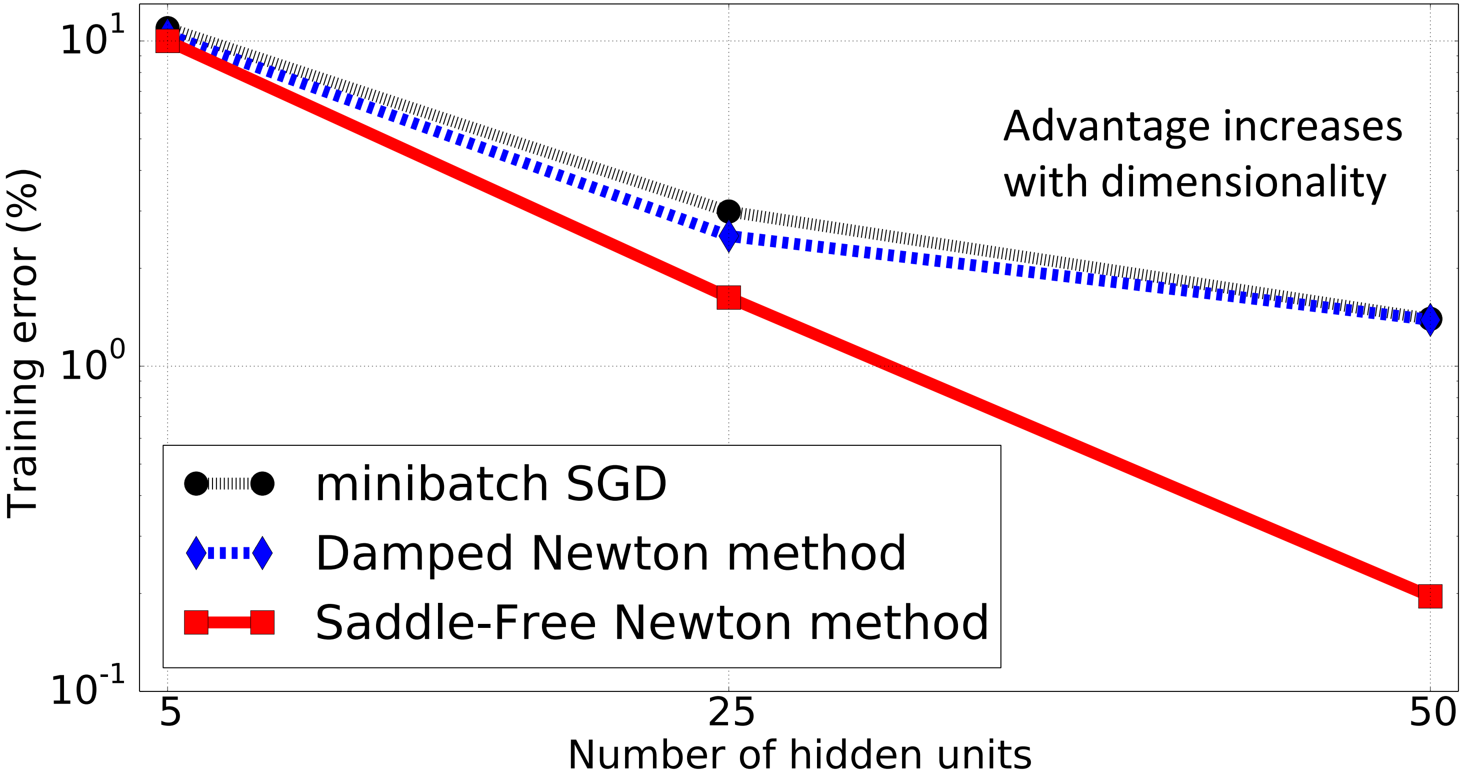
Shows that deep rectifier nets are analogous to spherical spin-glass models

The low-index critical points of large models concentrate in a band just above the global minimum



Saddle-Free Optimization (Pascanu, Dauphin, Ganguli, Bengio 2014)

- Saddle points are ATTRACTIVE for Newton's method
- Replace eigenvalues λ of Hessian by |λ|
- Justified as a particular trust region method



Other Priors That Work with Deep Distributed Representations

How do humans generalize from very few examples?

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

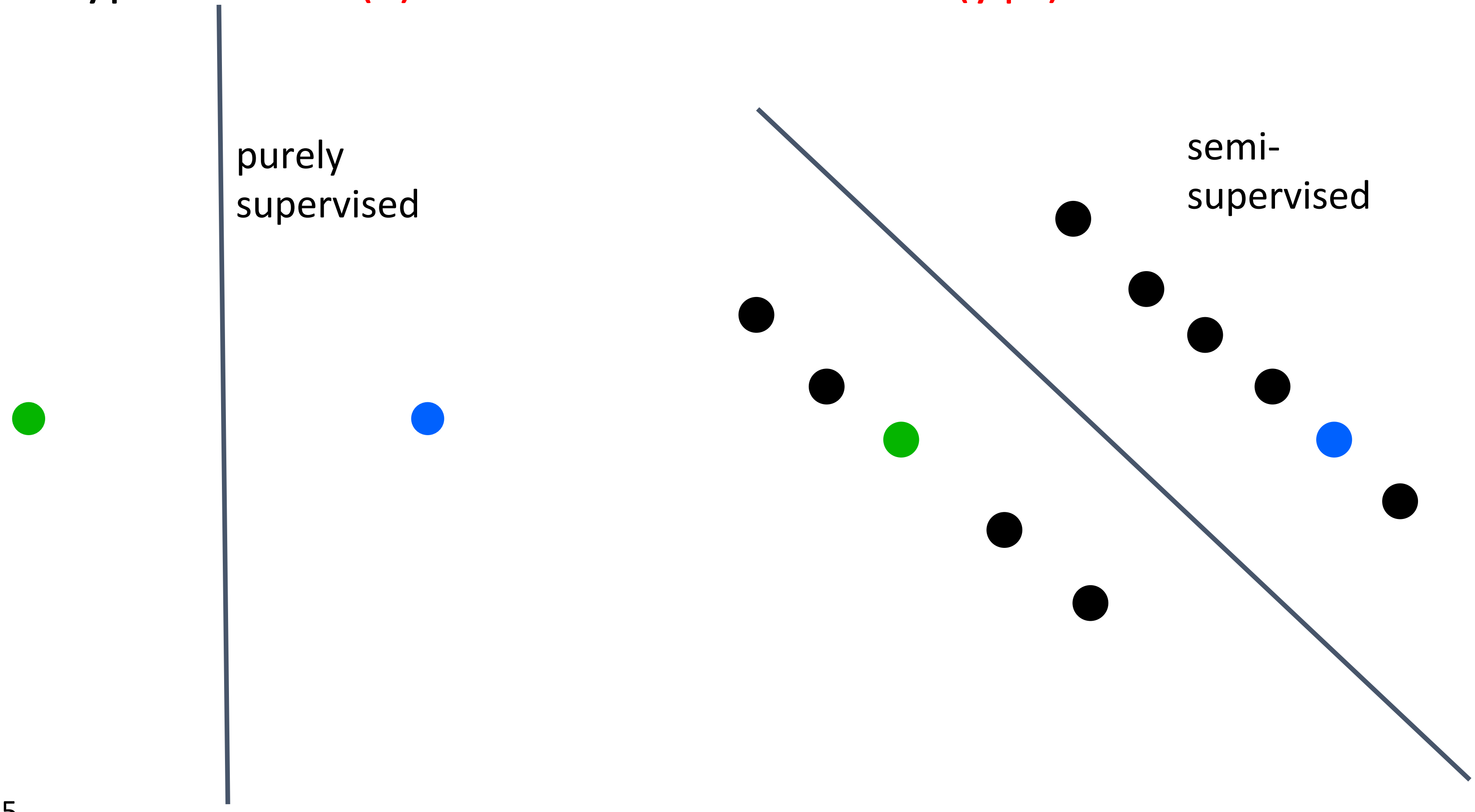
- Previous learning from: unlabeled data

  + labels for other tasks

- **Prior: shared underlying explanatory factors, in particular between P(x) and P(Y|x)**

Sharing Statistical Strength by Semi-Supervised Learning

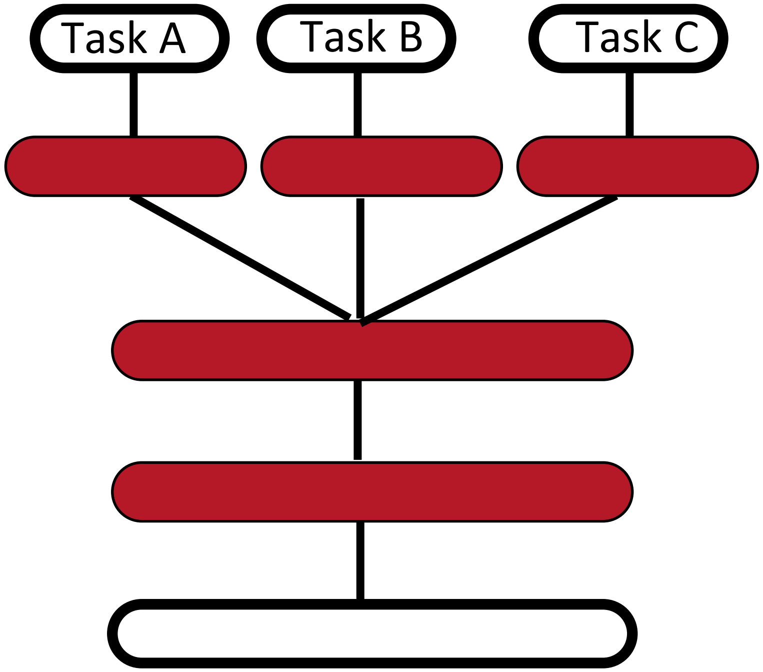
- Hypothesis: P(x) shares structure with P(y|x)



Multi-Task Learning

- Generalizing better to new tasks (tens of thousands!) is crucial to approach AI
- Deep architectures learn good intermediate representations that can be shared across tasks

  (Collobert & Weston ICML 2008, Bengio et al AISTATS 2011)
- Good representations that disentangle underlying factors of variation make sense for many tasks because **each task concerns a subset of the factors**



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

Prior: shared underlying explanatory factors between tasks

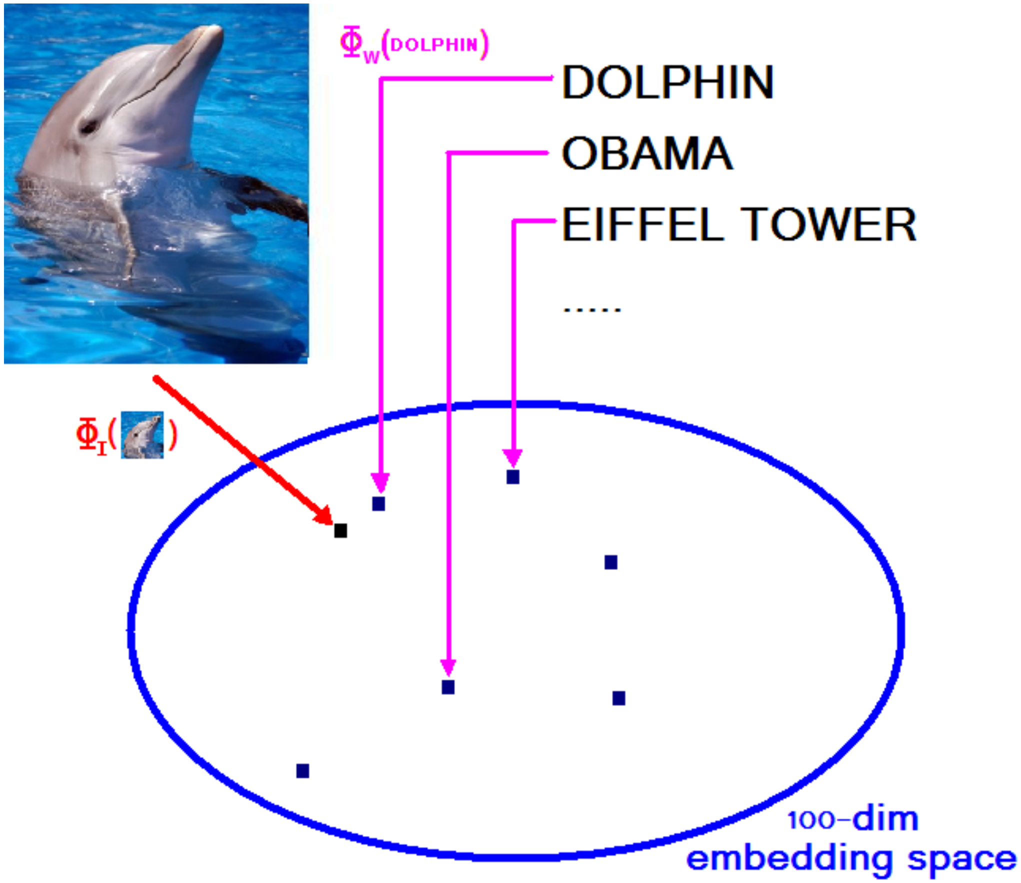
Google Image Search:

Different object types represented in the same space

Google:

S. Bengio, J. Weston & N. Usunier

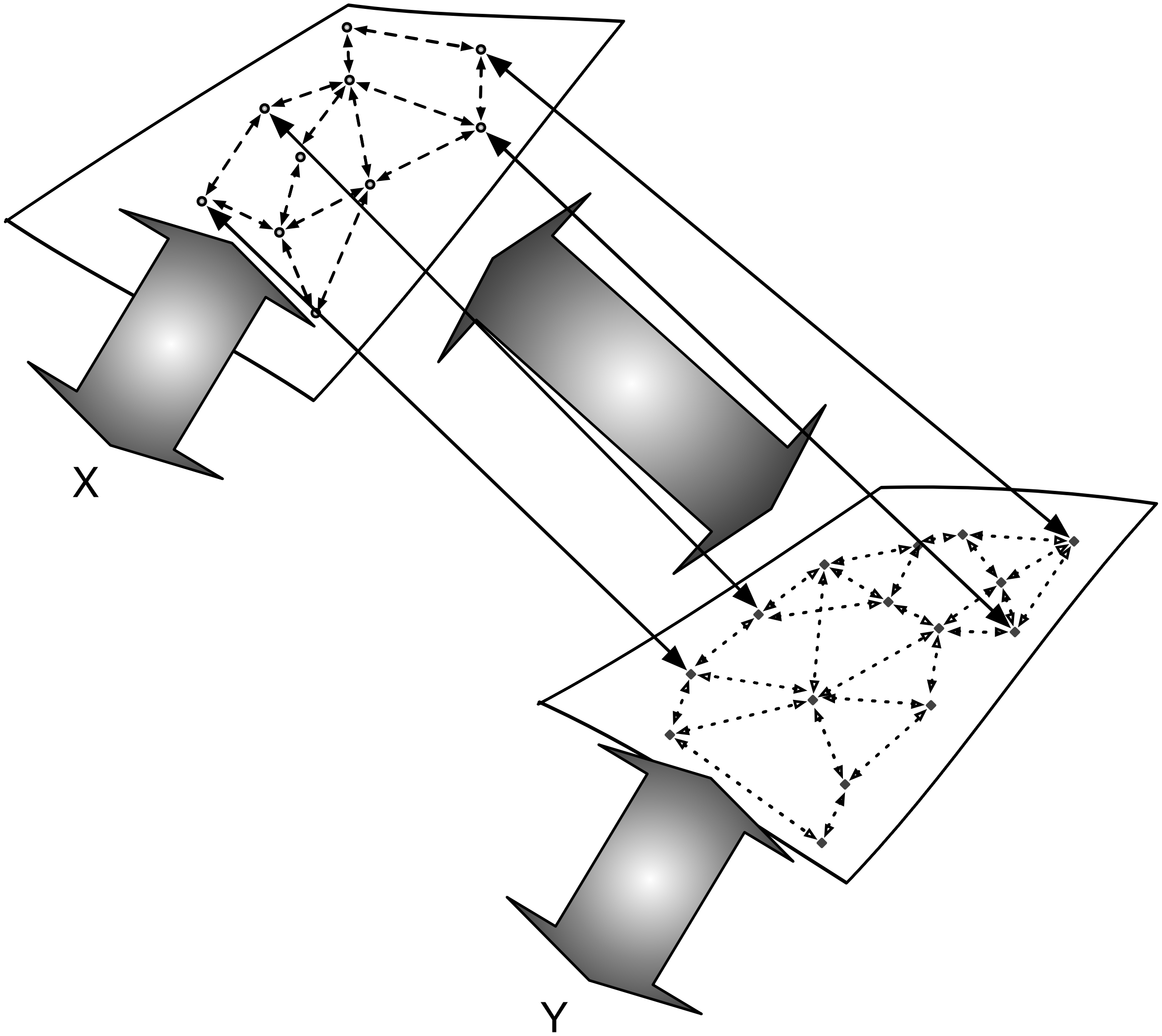
(IJCAI 2011, NIPS'2010, JMLR 2010, MLJ 2010)



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

Maps Between Representations

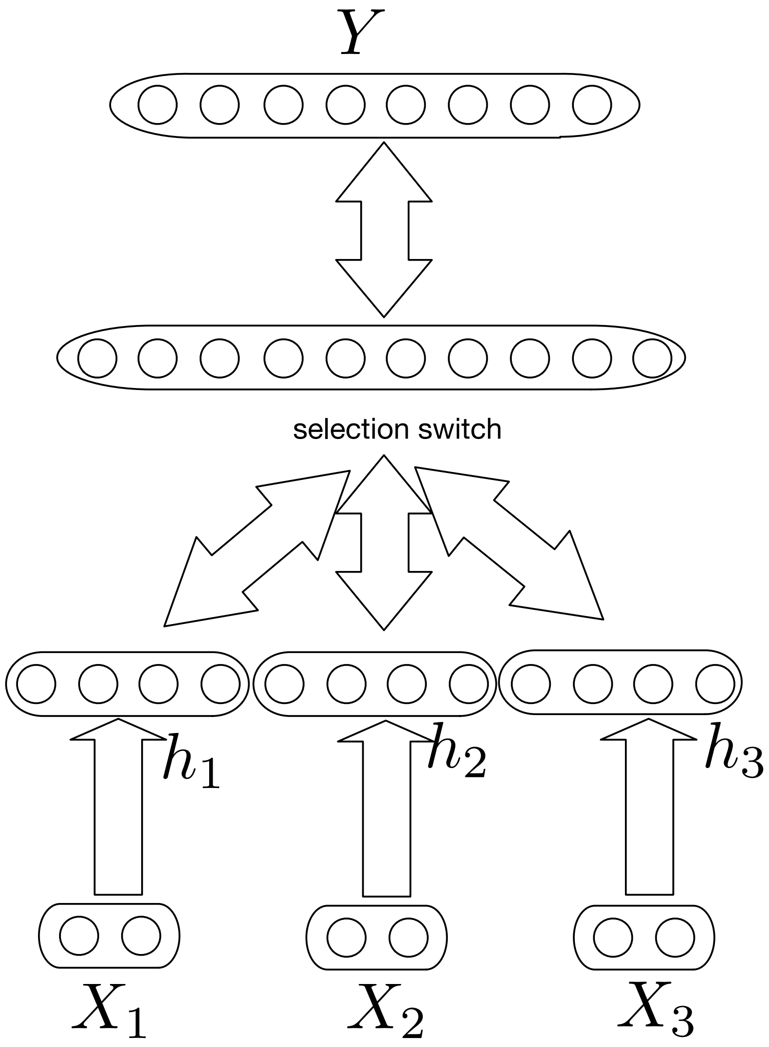
X and Y represent different modalities, e.g., image, text, sound…



Multi-Task Learning with Different Inputs for Different Tasks

E.g. speaker adaptation,

multi-modal input…



Why Latent Factors & Unsupervised Representation Learning? Because of Causality.

- If Ys of interest are among the causal factors of X, then

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

is tied to P(X) and P(X|Y), and P(X) is defined in terms of P(X|Y), i.e.

- The best possible model of X (unsupervised learning) MUST involve Y as a latent factor, implicitly or explicitly.
- Representation learning SEEKS the latent variables H that explain the variations of X, making it likely to also uncover Y.

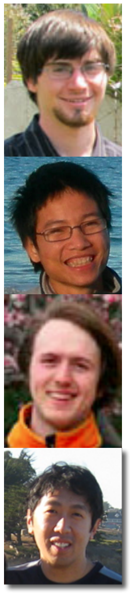
Invariance and Disentangling

- Invariant features

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

  avoid the curse of dimensionality

Emergence of Disentangling



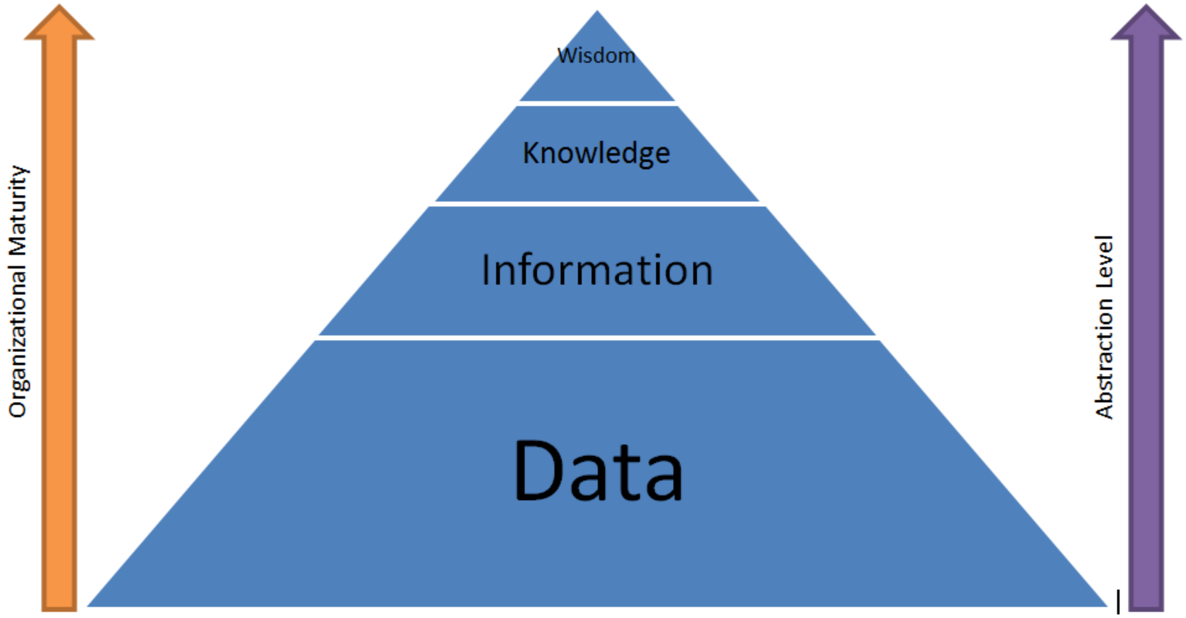
- (Goodfellow et al. 2009): sparse auto-encoders trained on images
  - some higher-level features more invariant to geometric factors of variation

- (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?

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



Conclusions

- **Distributed representations**:
  - prior that can buy exponential gain in generalization
- **Deep composition of non-linearities**:
  - prior that can buy exponential gain in generalization
- Both yield **non-local generalization**
- Strong evidence that **local minima are not an issue, saddle points**
- **Sharing factors = sharing statistical strengths:** semi-supervised learning, multi-task learning, multi-modal learning

MILA: Montreal Institute for Learning Algorithms