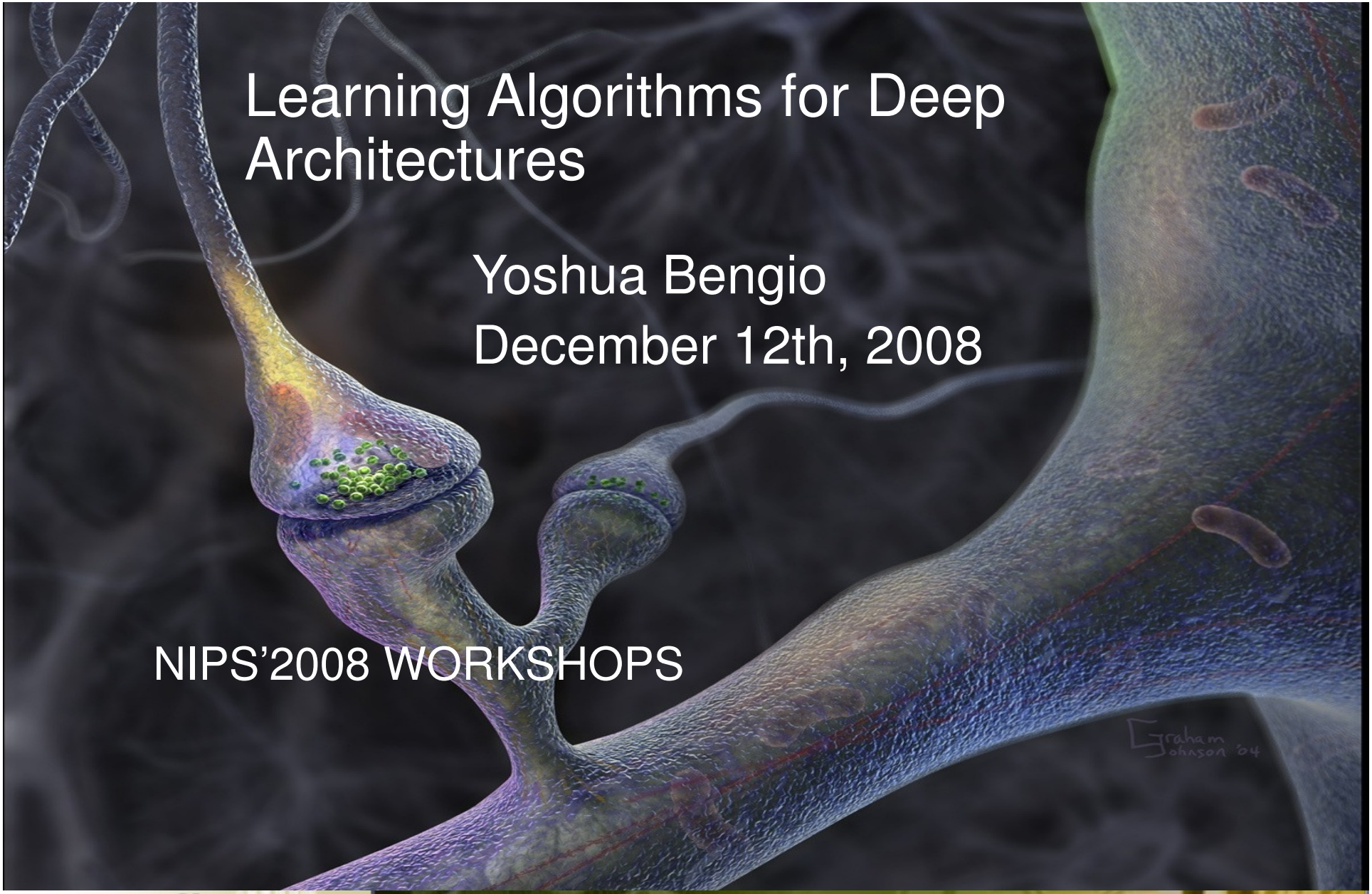


# Learning Algorithms for Deep Architectures

Yoshua Bengio

December 12th, 2008

NIPS'2008 WORKSHOPS

Université de Montréal

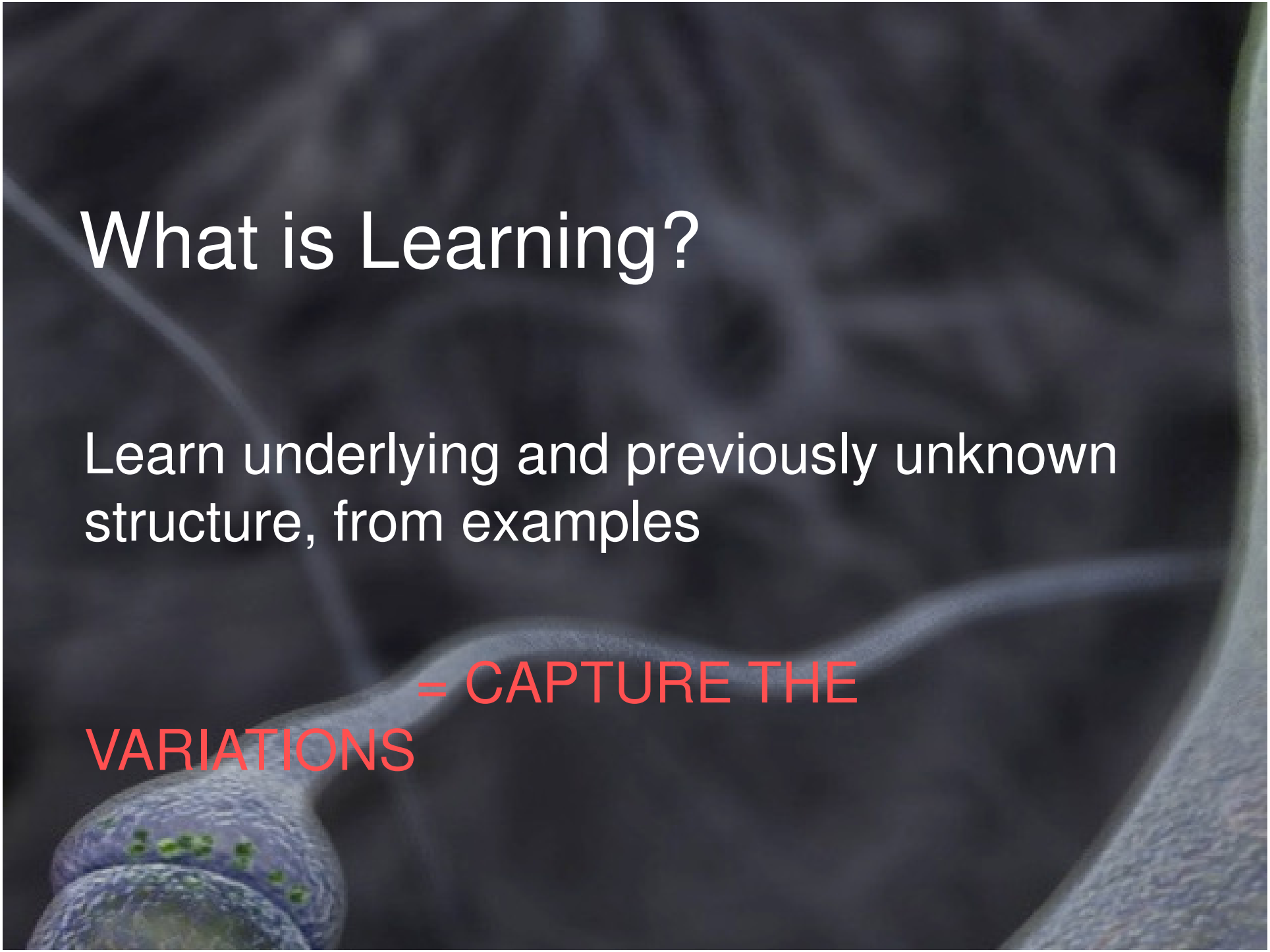
Département d'informatique et de recherche opérationnelle

# Local vs Distributed Representation

Debate since early 80's
(connectionist models)

Local representations:
- still common in neurosc.
- kernel machines, graphical models
- easier to interpret

Distributed representations:
- ≈ 1% active neurons in brains
- exponentially more efficient
- difficult optimization

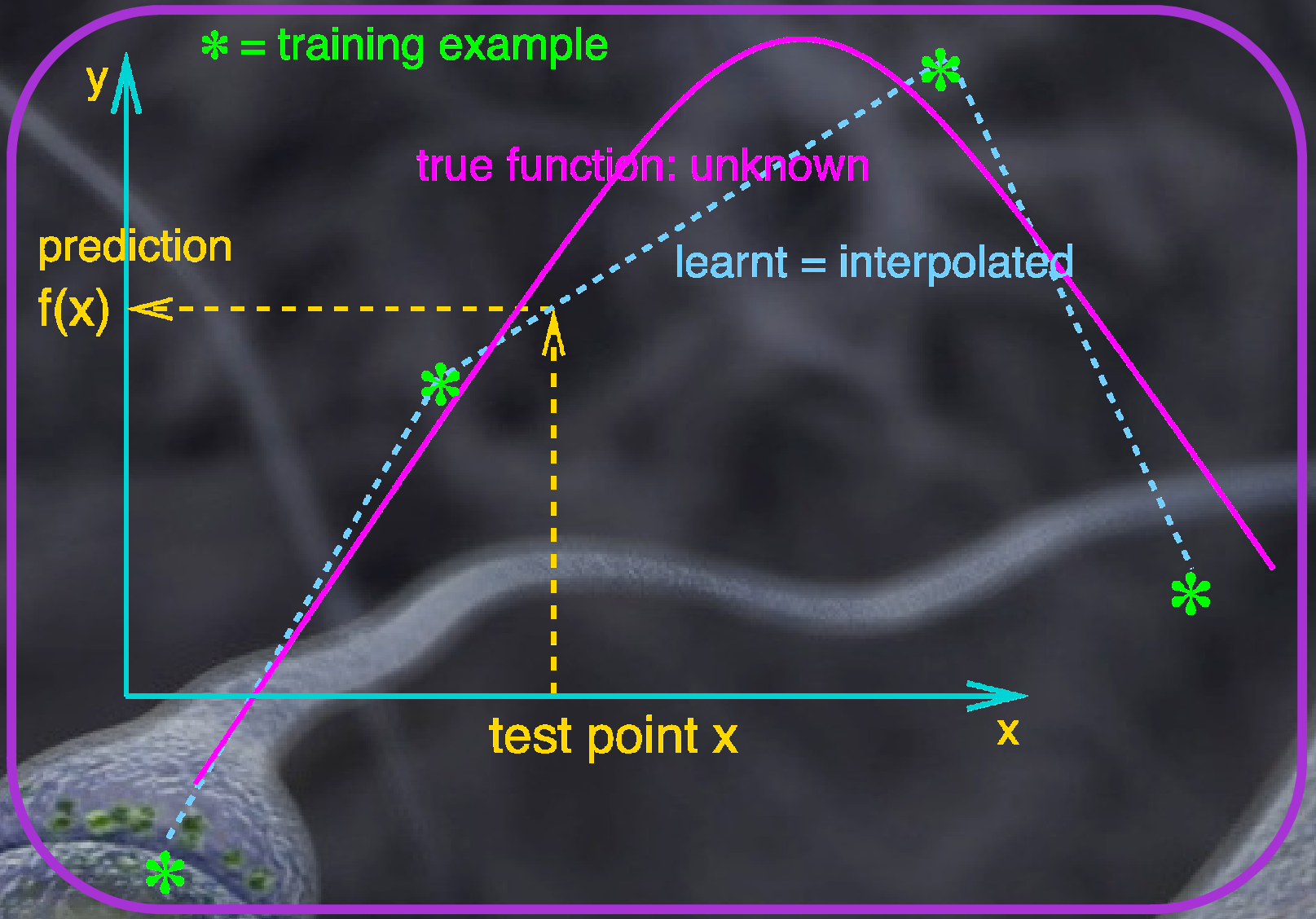


# What is Learning?

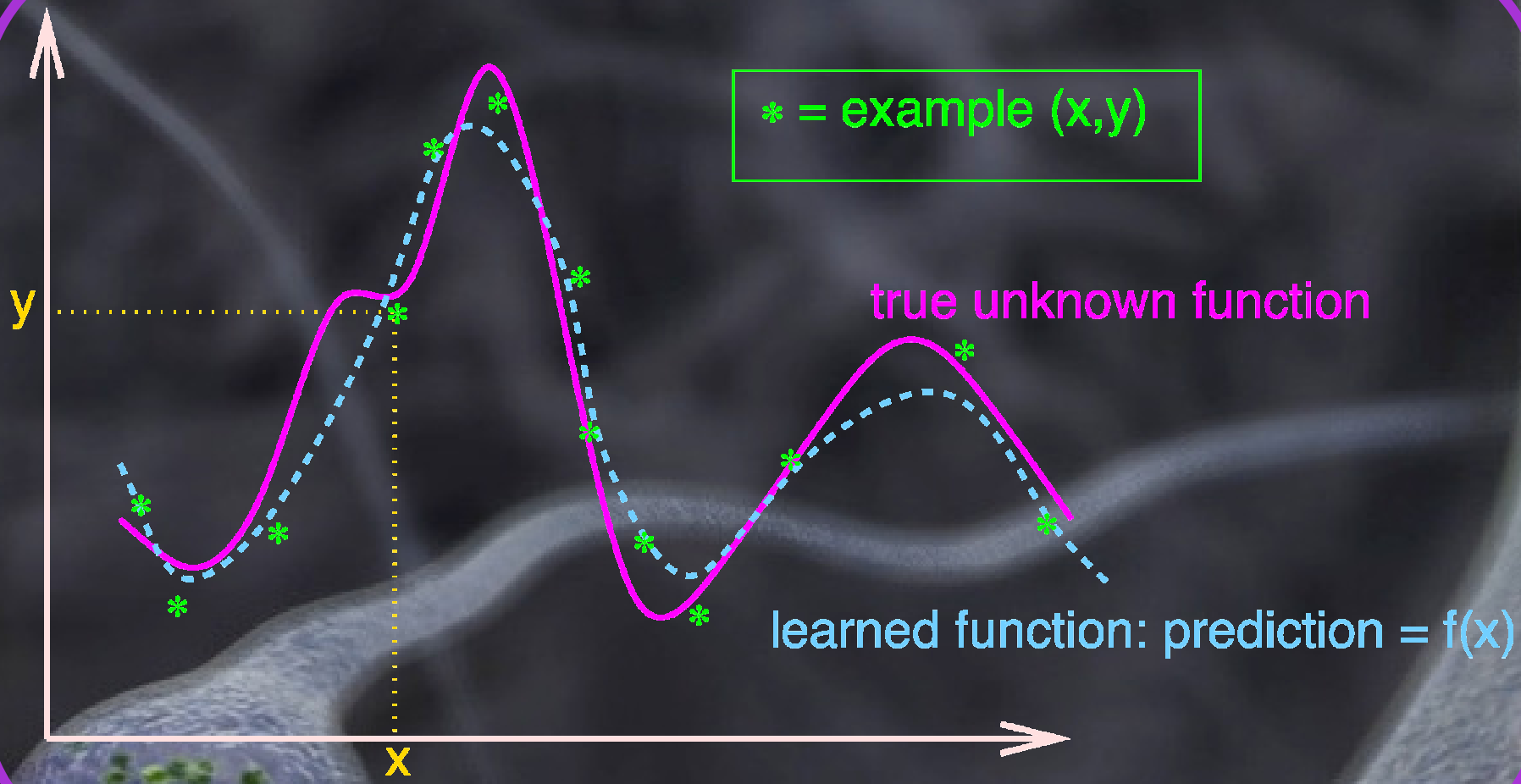
Learn underlying and previously unknown structure, from examples

= CAPTURE THE VARIATIONS

# Locally capture the variations

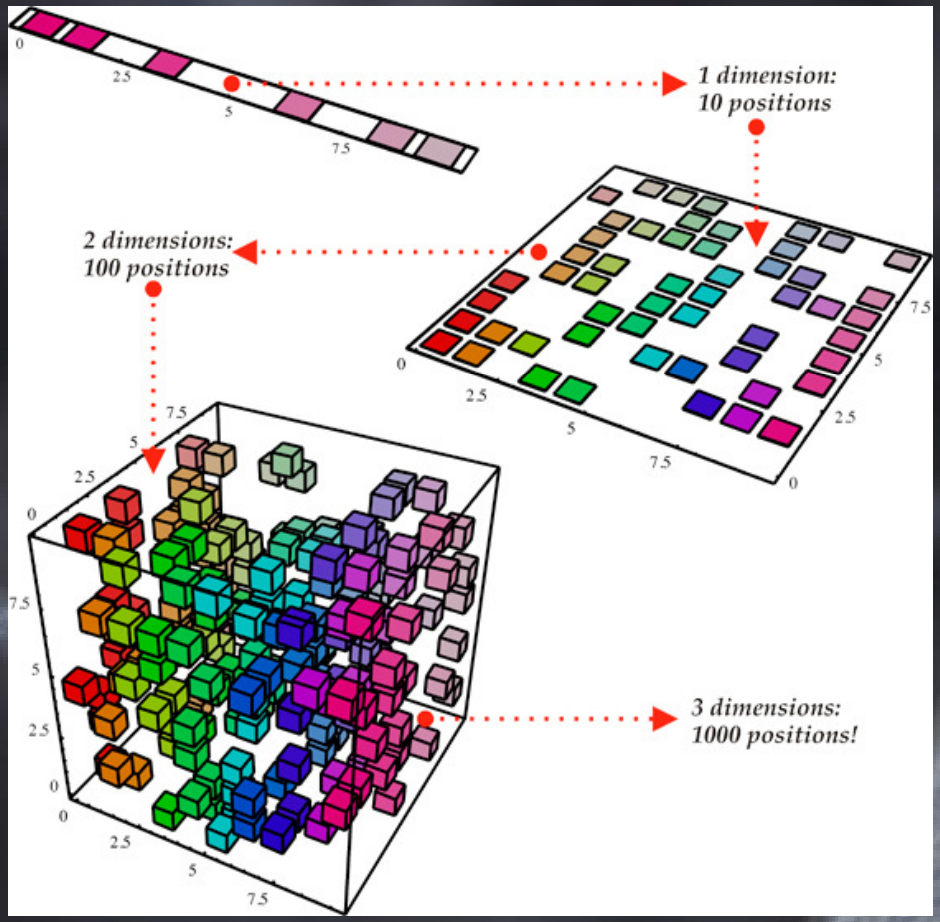


# Easy when there are only a few variations



# Curse of dimensionnality

To generalize locally, need examples representative of each possible variation.



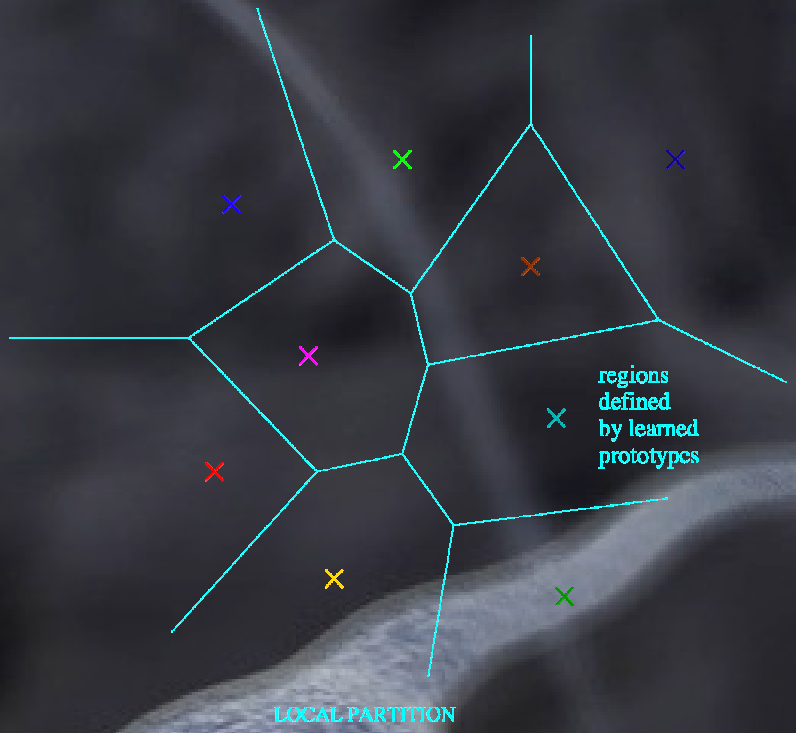
# Theoretical results

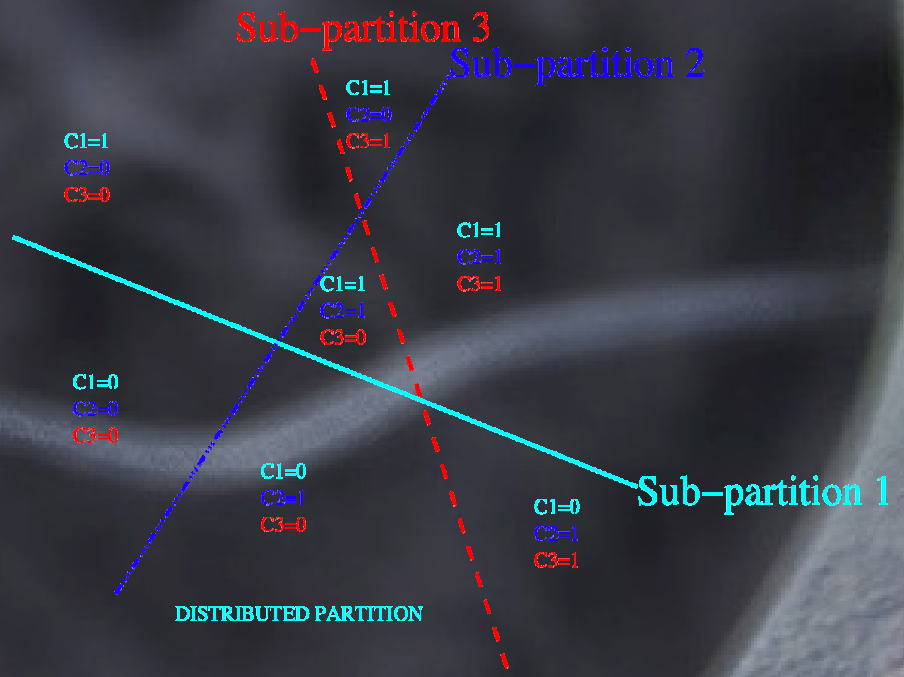
- **Theorem**: Gaussian kernel machines need at least k examples to learn a function that has 2k zero-crossings along some line

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

# Distributed Representations

Many neurons 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

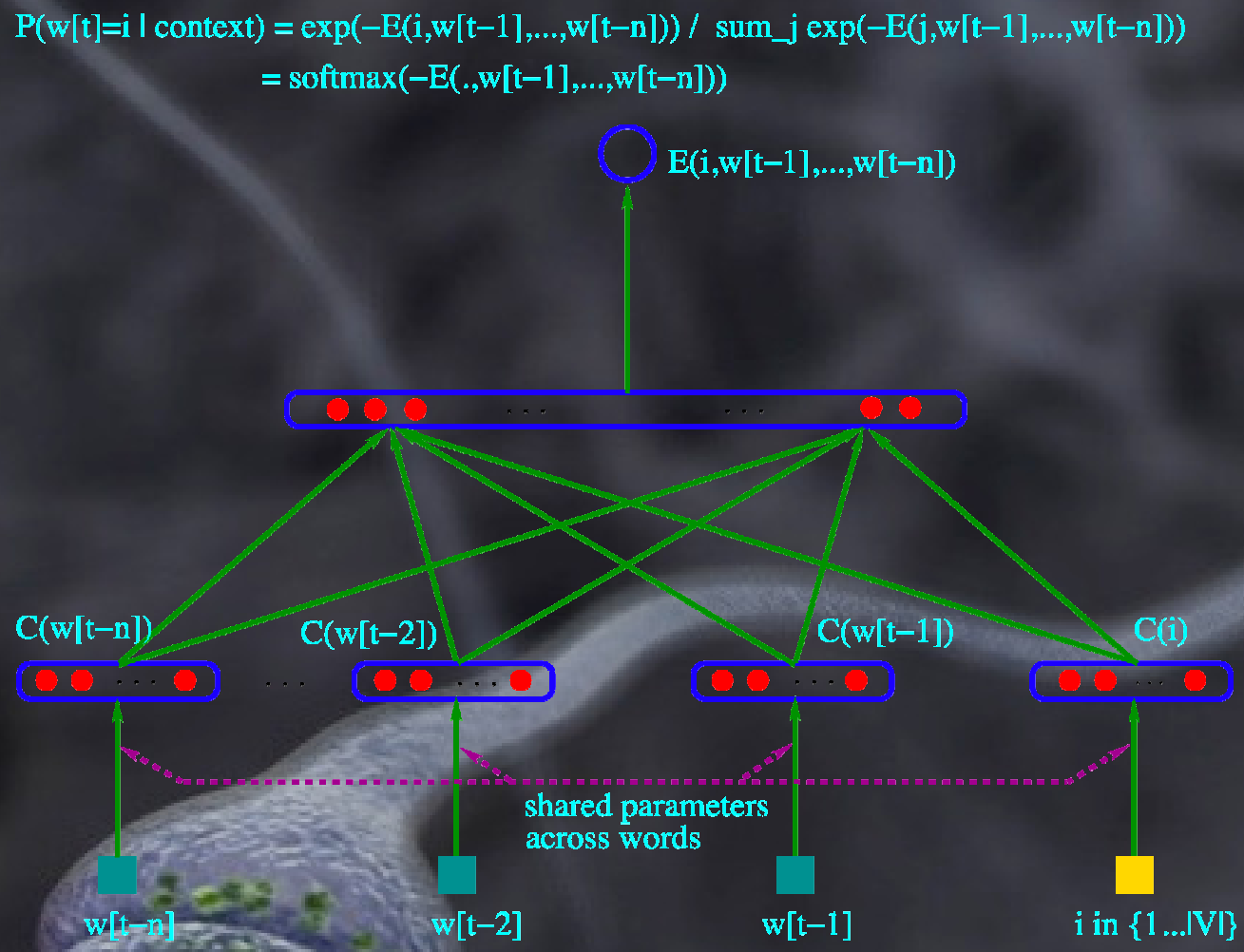


# Neurally Inspired Language Models

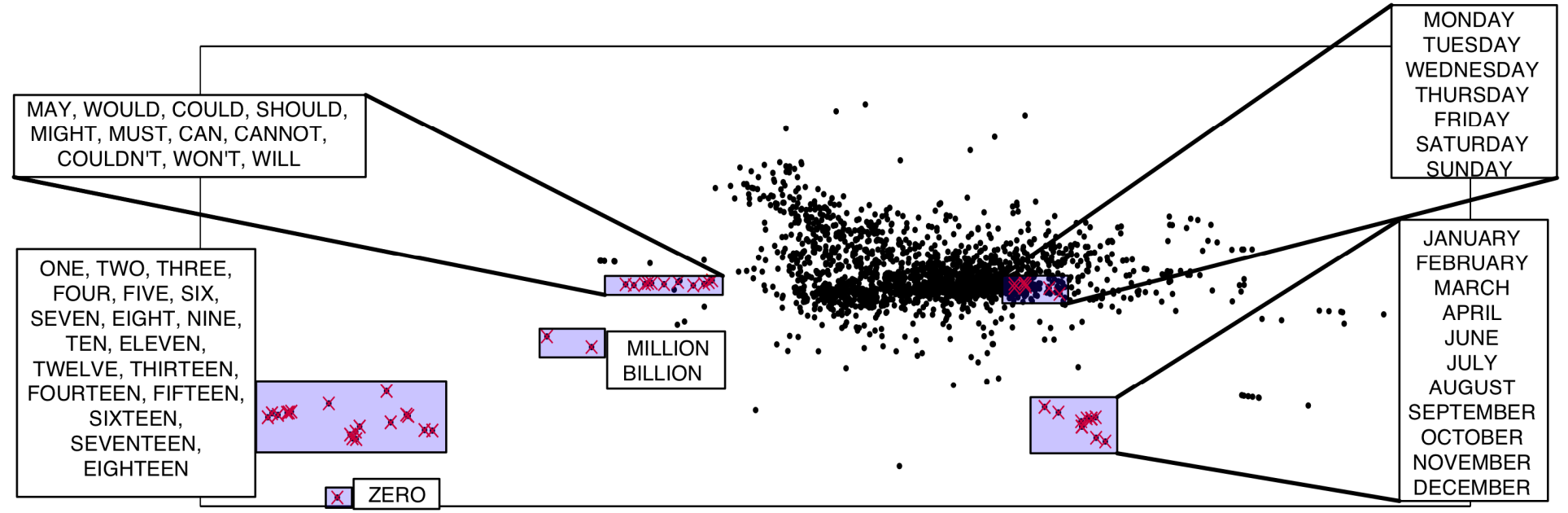
- Classical statistical models of word sequences: local representations
- Input = sequence of symbols, each element of sequence = 1 of N possible words
- Distributed representations: learn to embed the words in a continuous-valued low-dimensional semantic space

# Neural Probabilistic Language Models




Successes of this architecture and its descendents: beats localist state-of-the-art in NLP in most tasks (language model, chunking, semantic role labeling, POS)

# Embedding Symbols

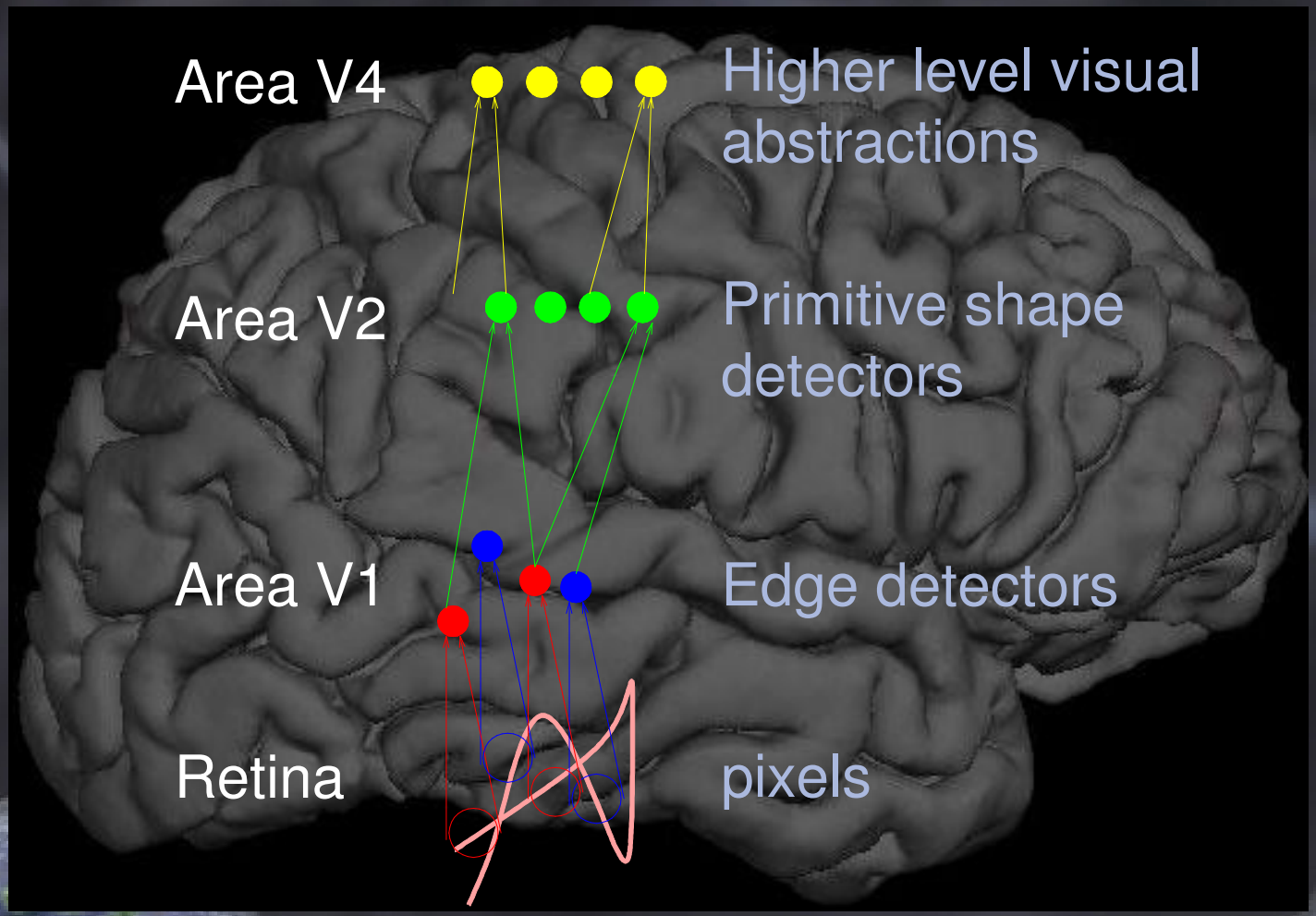


# Nearby Words in Semantic Space

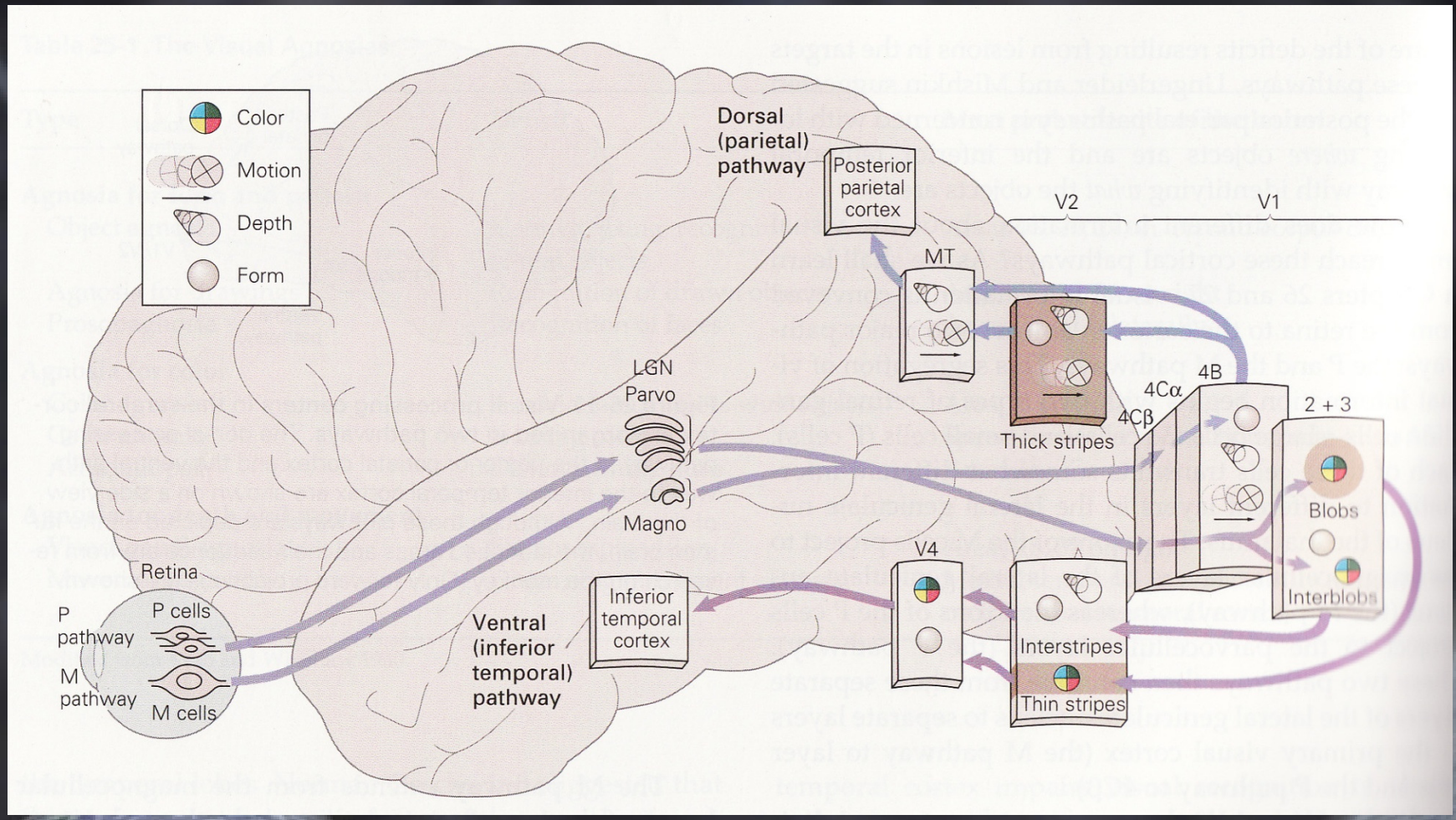
| France | Jesus | XBOX | Reddish | Scratched |
|---|---|---|---|---|
| Spain | Christ | Playstation | Yellowish | Smashed |
| Italy | God | Dreamcast | Greenish | Ripped |
| Russia | Resurrection | PS### | Brownish | Brushed |
| Poland | Prayer | SNES | Bluish | Hurled |
| England | Yahweh | WH | Creamy | Grabbed |
| Denmark | Josephus | NES | Whitish | Tossed |
| Germany | Moses | Nintendo | Blackish | Squeezed |
| Portugal | Sin | Gamecube | Silvery | Blasted |
| Sweden | Heaven | PSP | Greyish | Tangled |
| Austria | Salvation | Amiga | Paler | Slashed |

Collobert & Weston, ICML'2008

# Deep Architecture in the Brain



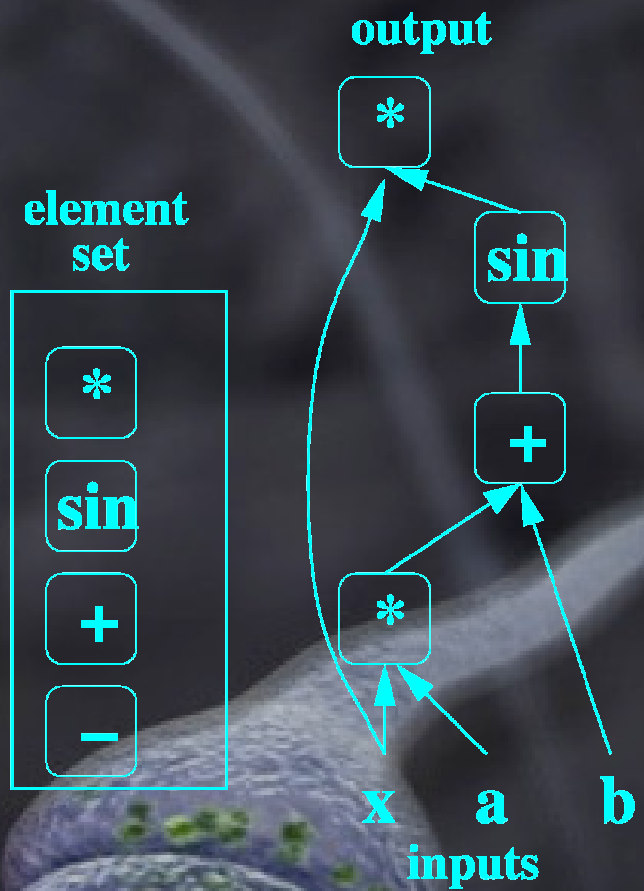
# Visual System




**Sequence of transformations / abstraction levels**

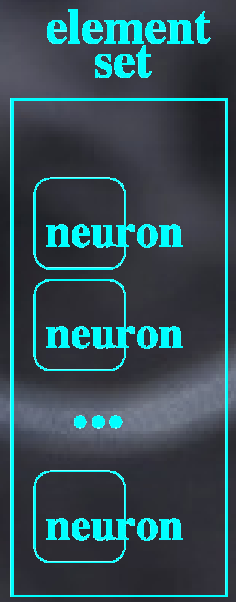
# Architecture Depth

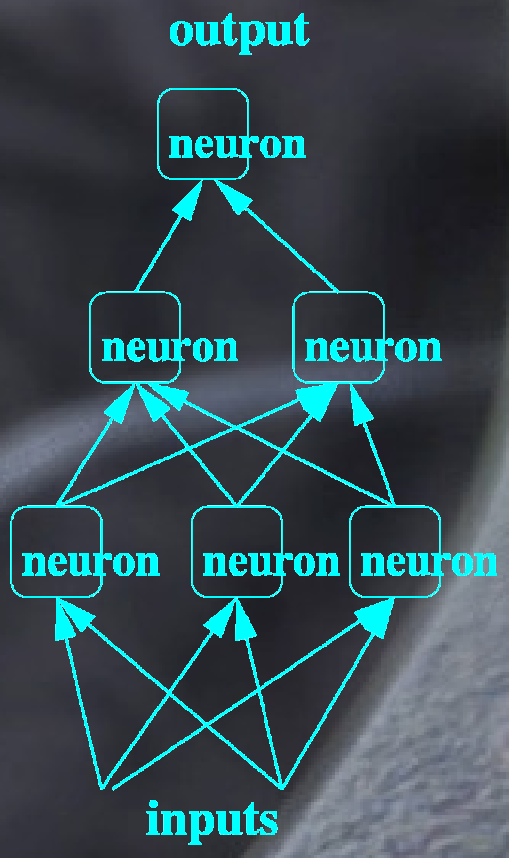
Computation performed by learned function can be decomposed into a graph of simpler operations







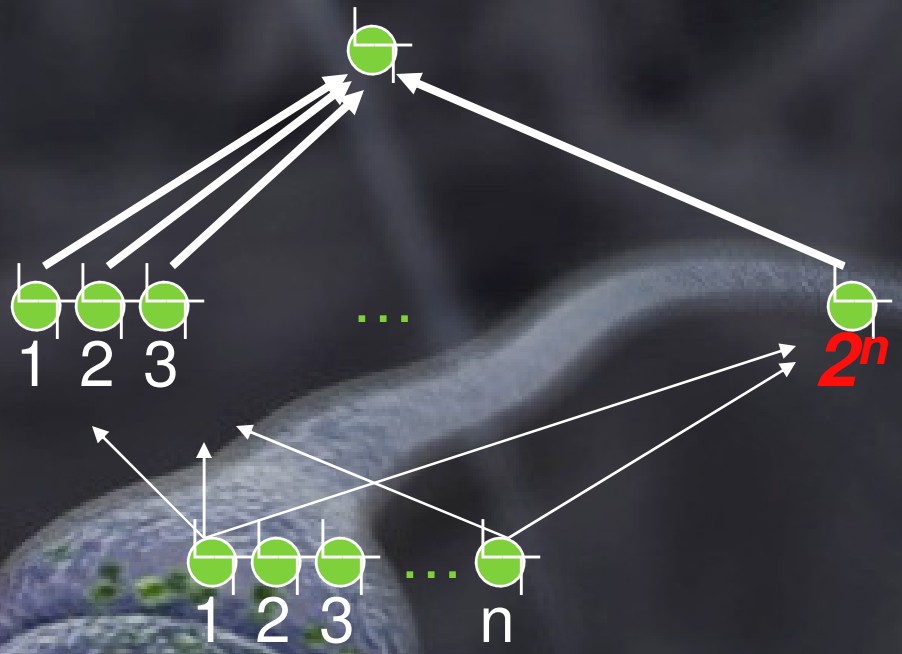


# Insufficient Depth

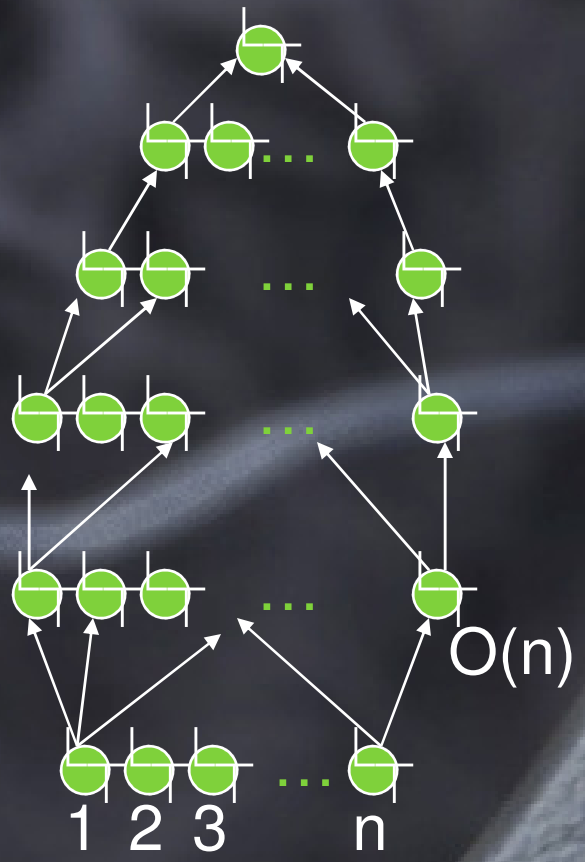
**Insufficient depth =**

**May require exponential-size architecture**




**Sufficient depth =**

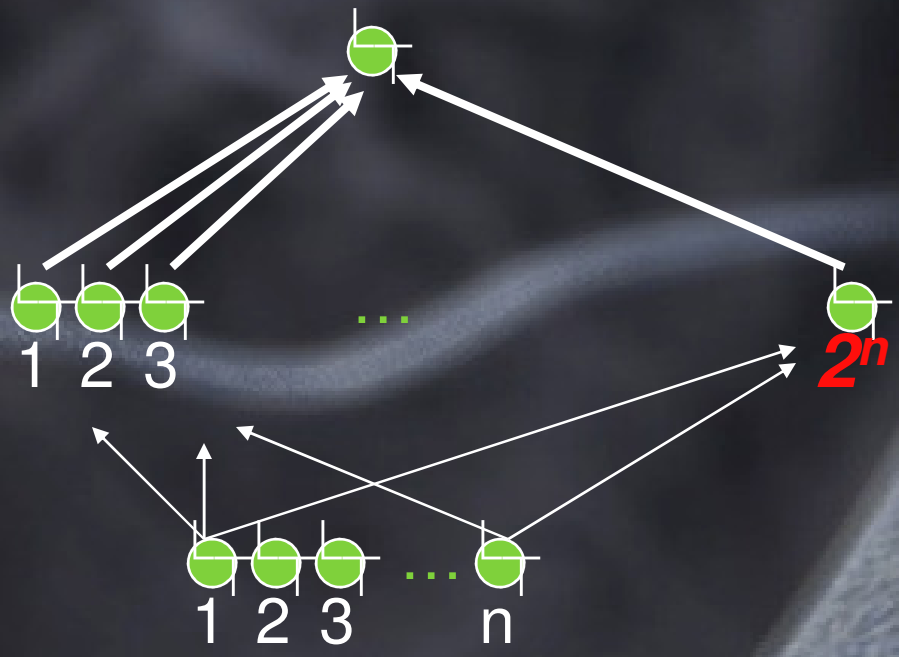
**Compact representation**



# Good News, Bad News

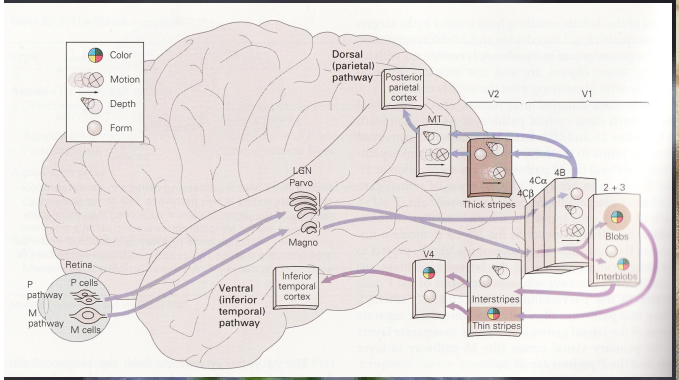
2 layers of { logic gates / formal neurons / RBF units } = universal approximator

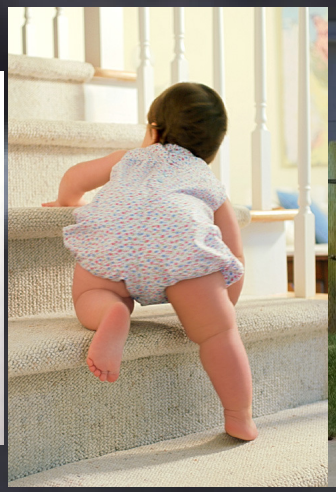
Theorems for all 3:
(Hastad et al 86 & 91, Bengio et al 2007)
Functions representable compactly with k layers may require exponential size with k-1 layers

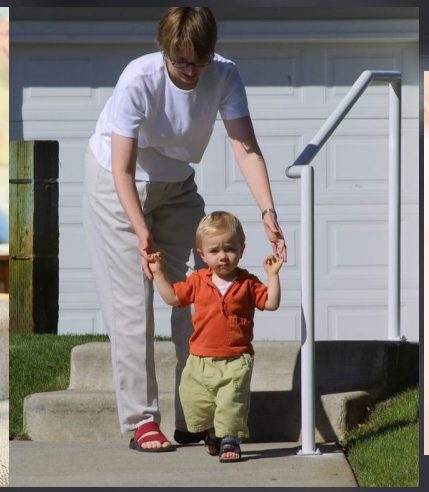


# Neuro-cognitive inspiration

- Brains use a distributed representation
- Brains use a deep architecture
- Brains heavily use unsupervised learning
- Brains learn simpler tasks first
- Human brains developed with society / culture / education





# Breakthrough!

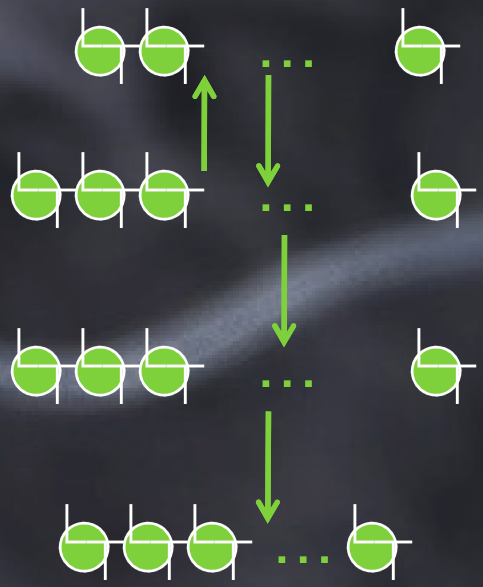
Before 2006

Failure of deep architectures

After 2006

Train one level after the other, unsupervised, extracting abstractions of gradually higher level

Deep Belief Networks (Hinton et al 2006)

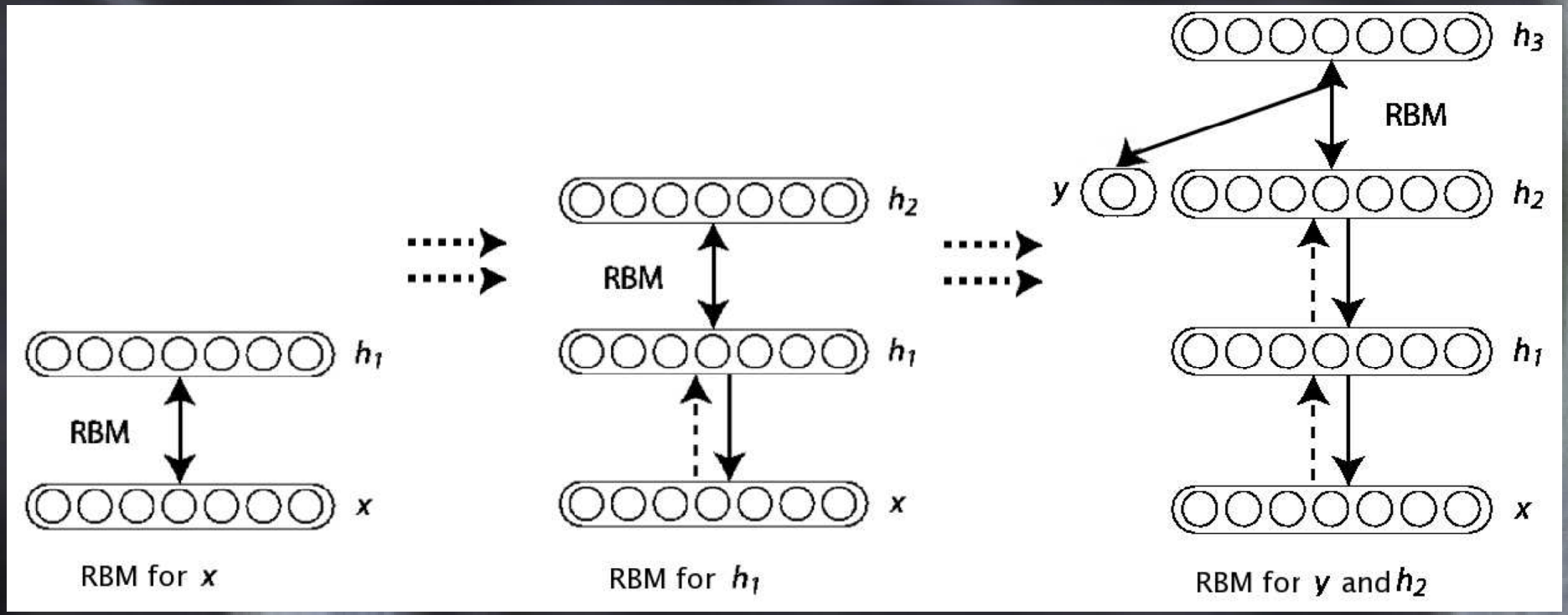


# Success of deep distributed neural networks

Since 2006

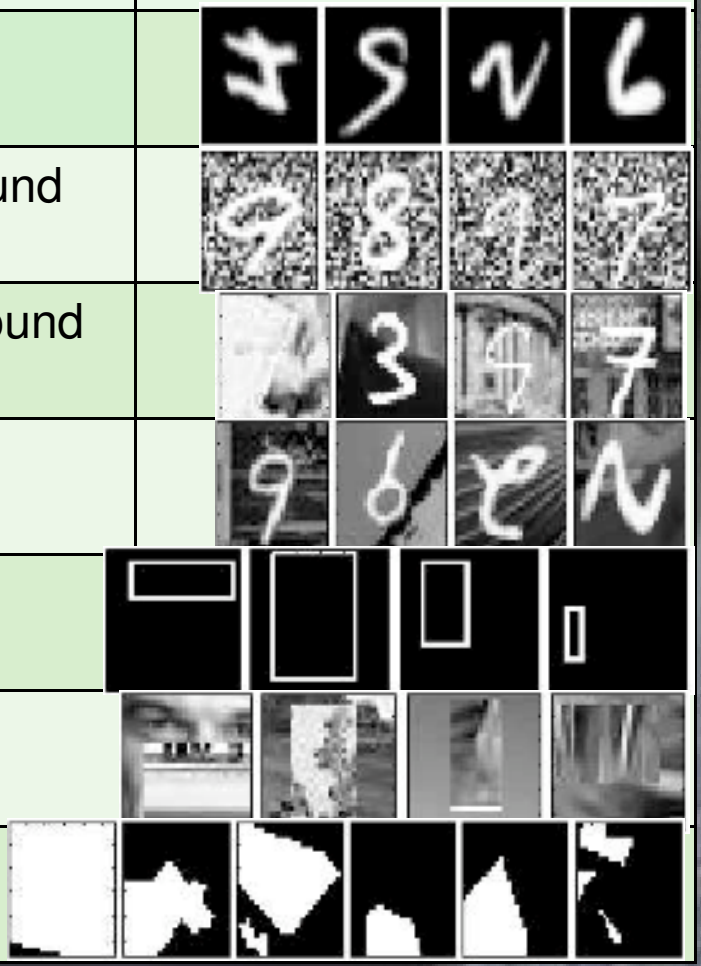
- Records broken on MNIST handwritten character recognition benchmark
- State-of-the-art beaten in language modeling (Collobert & Weston 2008)
- NSF et DARPA are interested…
- Similarities between V1 & V2 neurons and representations learned with deep nets
- (Raina et al 2008)

# Unsupervised greedy layer-wise pre-training



# Image Data Sets

| | | |
|---|---|---|
| **Basic** | 10k-example subset of MNIST | |
| **Rot** | Rotated digits | |
| **Bg-rand** | Digits on top of white noise background | |
| **Bg-img** | Digits on top of image patch background | |
| **Rot-bg-img** | Rotated digits on image background | |
| **Rect** | Distinguish tall vs wide rectangles | |
| **Rect-img** | Rectangles on image background | |
| **Convex** | Is the polygon convex? | |

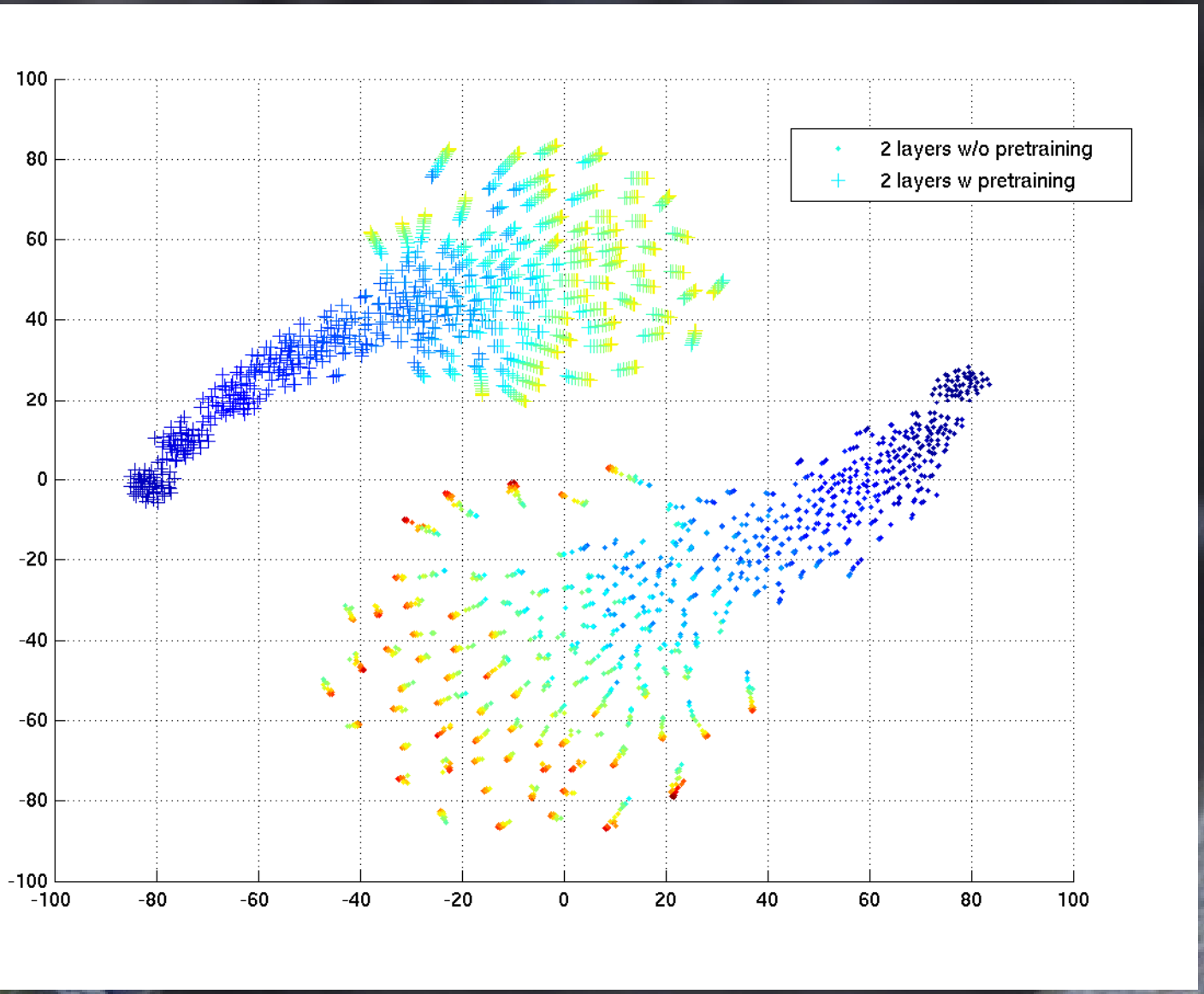


| Problem | $SVM_{rbf}$ | DBN-1 | DBN-3 | SAA-3 | SdA-3 ($\nu$) | $SVM_{rbf}(\nu)$ |
|---|---|---|---|---|---|---|
| basic | 3.03±0.15 | 3.94±0.17 | 3.11±0.15 | 3.46±0.16 | 2.80±0.14 (10%) | 3.07 (10%) |
| rot | 11.11±0.28 | 14.69±0.31 | 10.30±0.27 | 10.30±0.27 | 10.29±0.27 (10%) | 11.62 (10%) |
| bg-rand | 14.58±0.31 | 9.80±0.26 | 6.73±0.22 | 11.28±0.28 | 10.38±0.27 (40%) | 15.63 (25%) |
| bg-img | 22.61±0.37 | 16.15±0.32 | 16.31±0.32 | 23.00±0.37 | 16.68±0.33 (25%) | 23.15 (25%) |
| rot-bg-img | 55.18±0.44 | 52.21±0.44 | 47.39±0.44 | 51.93±0.44 | 44.49±0.44 (25%) | 54.16 (10%) |
| rect | 2.15±0.13 | 4.71±0.19 | 2.60±0.14 | 2.41±0.13 | 1.99±0.12 (10%) | 2.45 (25%) |
| rect-img | 24.04±0.37 | 23.69±0.37 | 22.50±0.37 | 24.05±0.37 | 21.59±0.36 (25%) | 23.00 (10%) |
| convex | 19.13±0.34 | 19.92±0.35 | 18.63±0.34 | 18.41±0.34 | 19.06±0.34 (10%) | 24.20 (10%) |

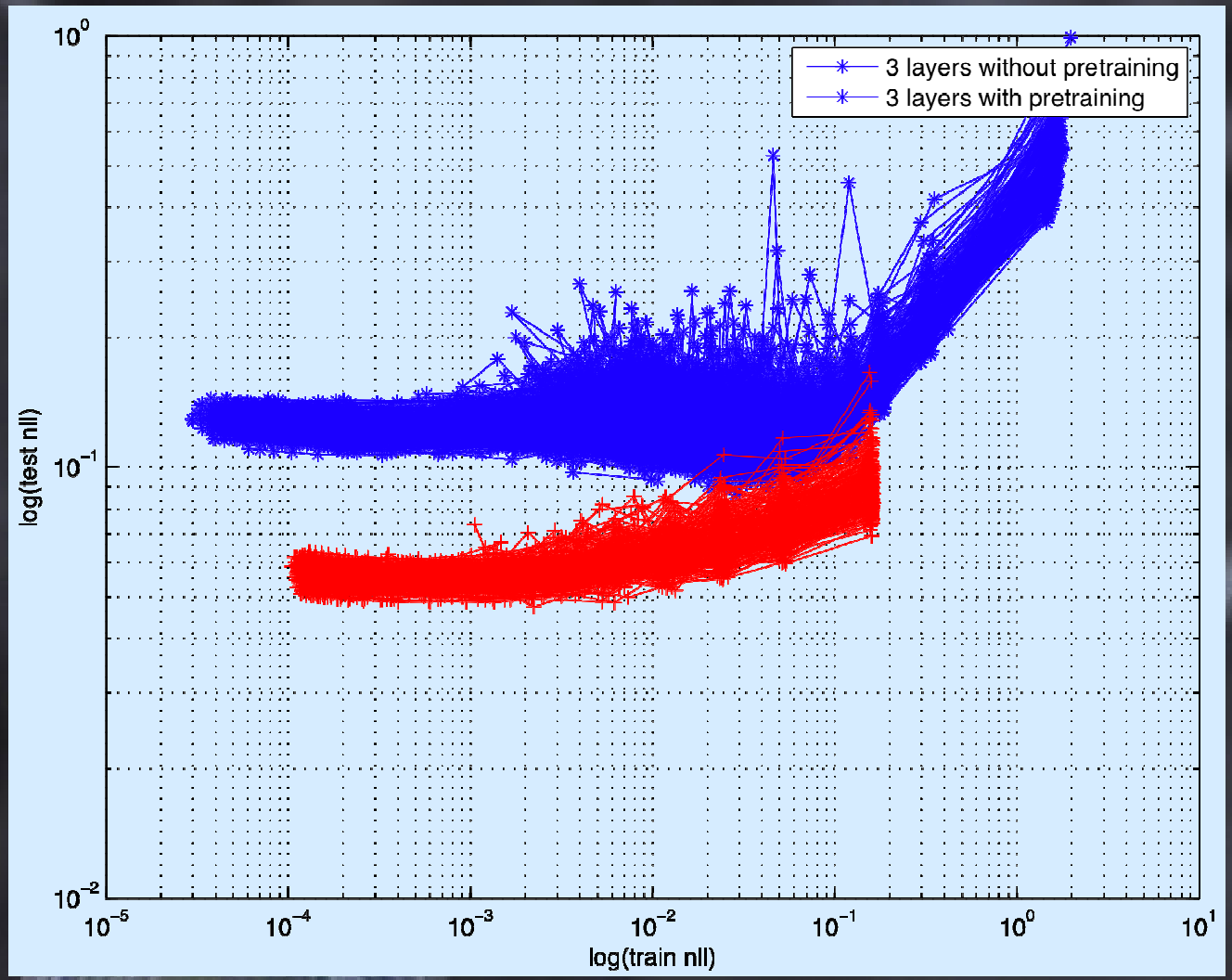
# Why is unsupervised pre-training working?

- Learning can be mostly local with unsupervised learning of transformations (Bengio 2008)
- generalizing better in presence of many factors of variation (Larochelle et al ICML'2007)
- deep neural nets iterative training: stuck in poor local minima (AISTATS submission)
- pre-training moves into improbable region with better basins of attraction
- Training one layer after the other ≈ continuation method (Bengio 2008)

# Flower Power



# Unsupervised pre-training acts as a regularizer




- Lower test error at same training error
- Hurts when capacity is too small
- Preference for transformations capturing input distribution, instead of w=0
- But helps to optimize lower layers.

# Non-convex optimization

- Humans somehow find a good solution to an intractable non-convex optimization problem. How?
  - Shaping? The order of examples / stages in development / education

≈ approximate global optimization (continuation)

# Continuation methods

First learn simpler tasks, then build on top and learn higher-level abstractions.

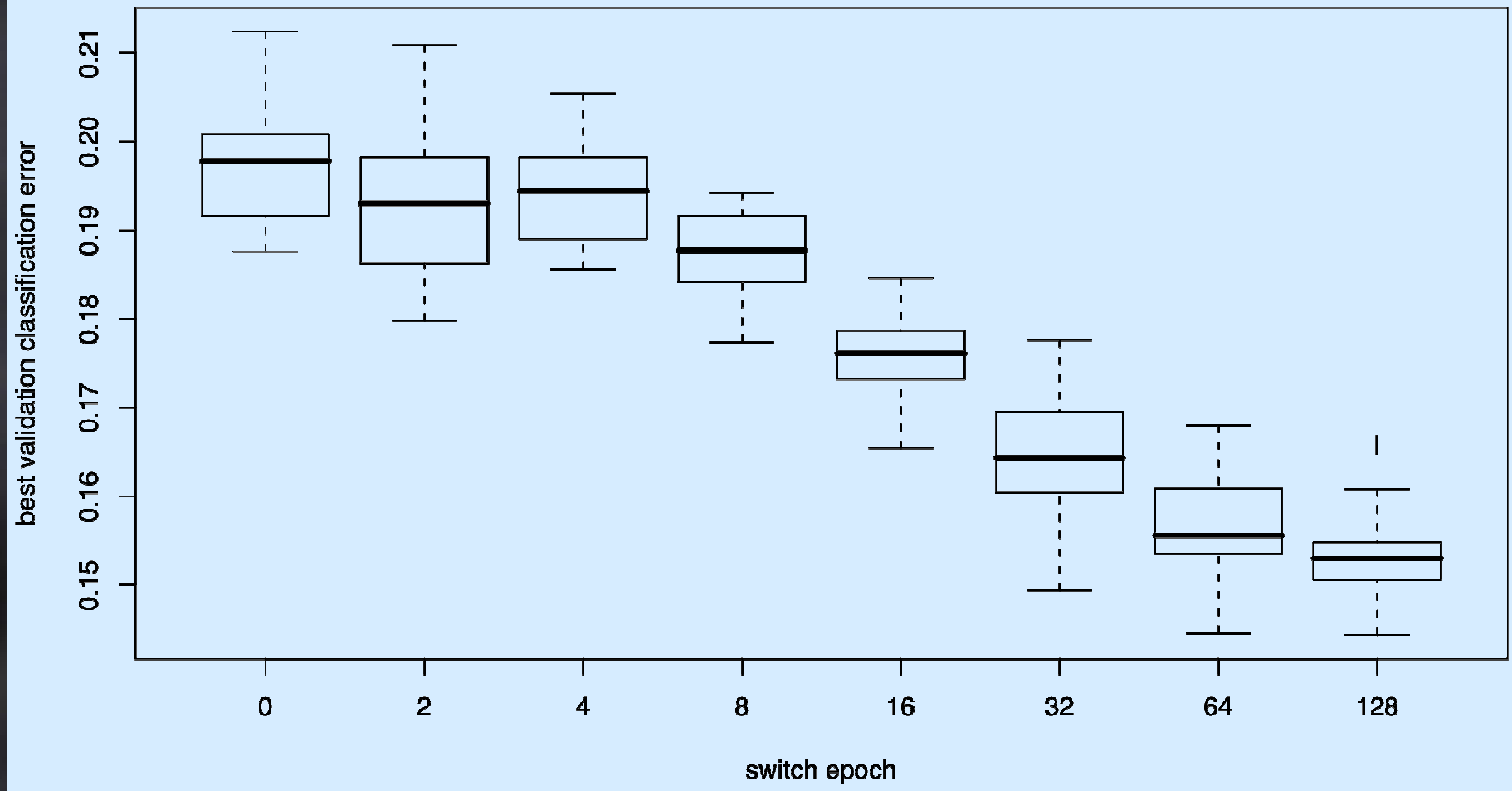
# Experiments on multi-stage curriculum training

Stage 1 data:

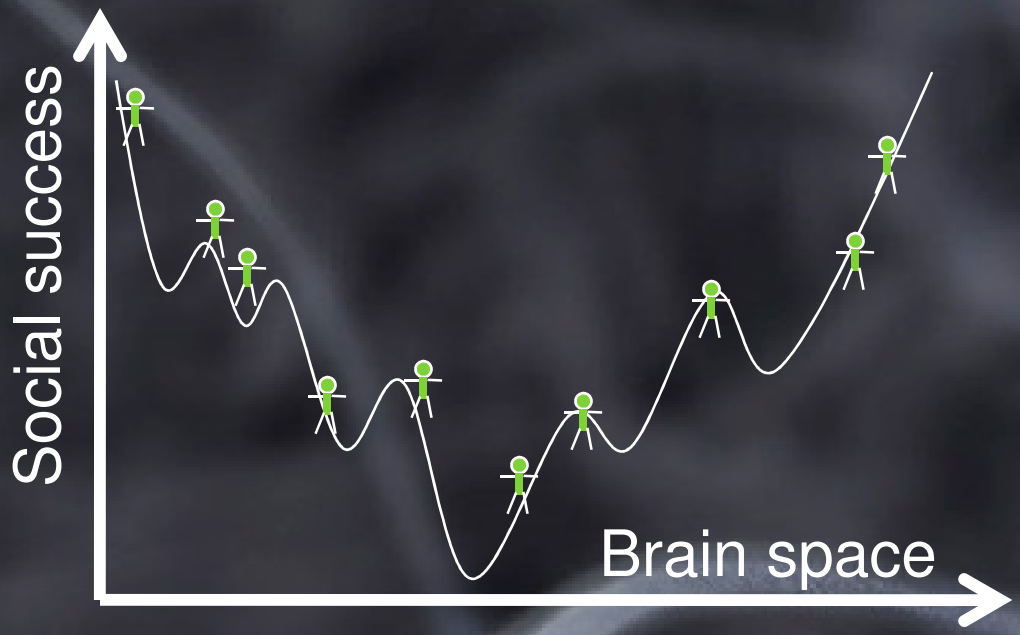
Stage 2: data

Train deep net for 128 epochs. Switch from stage 1 to stage 2 data at epoch N in {0, 2, 4, 8, 16, 32, 64, 128}.

# The wrong distribution helps



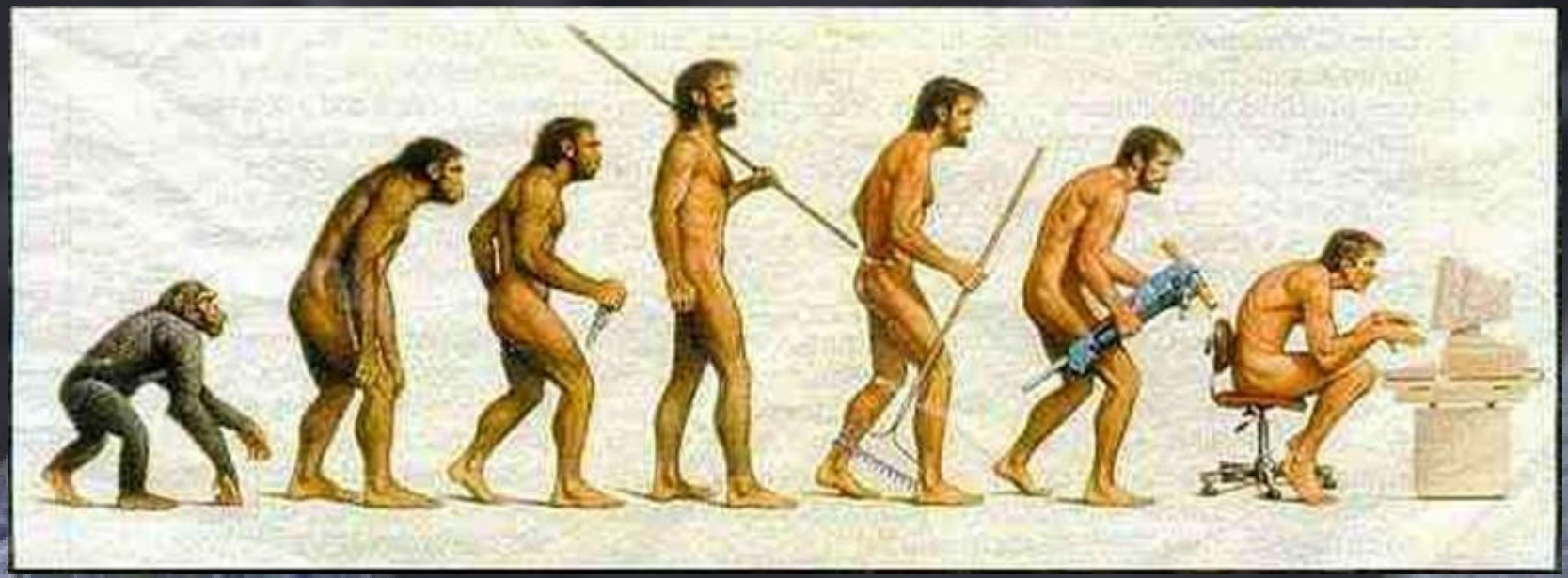
# Parallelized exploration: Evolution of concepts




- Each brain explores a different potential solution
- Instead of exchanging synaptic configurations, exchange ideas through language

# Evolution of concepts: memes

- Genetic algorithms need 2 ingredients:
  - Population of candidate solutions: brains
  - Recombination mechanism: culture/language



# Conclusions

1. Representation: brain-inspired & distributed
2. Architecture: brain-inspired & deep
    1. Challenge: non-convex optimization
    2. Plan: understand the issues and try to view what brains do as strategies for solving this challenge