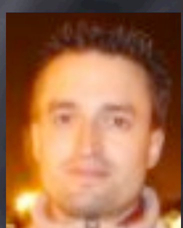
Learning Algorithms for Deep Architectures

Yoshua Bengio

December 12th, 2008

NIPS'2008 WORKSHOPS



Olivier Delalleau

Joseph Turian

Dumitru Erhan

Pierre-Antoine Manzagol

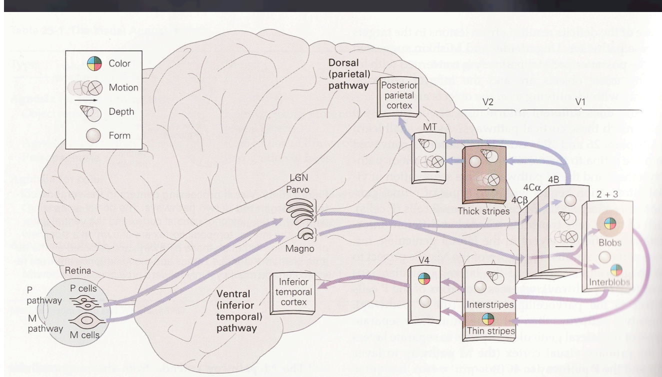
Jérôme Louradour

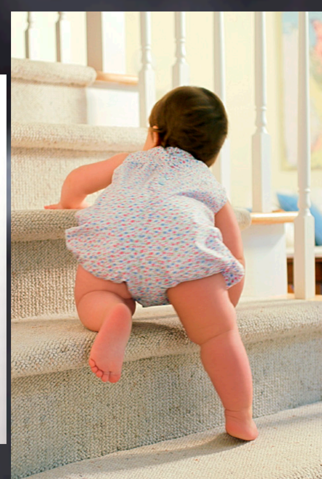
Université de Montréal

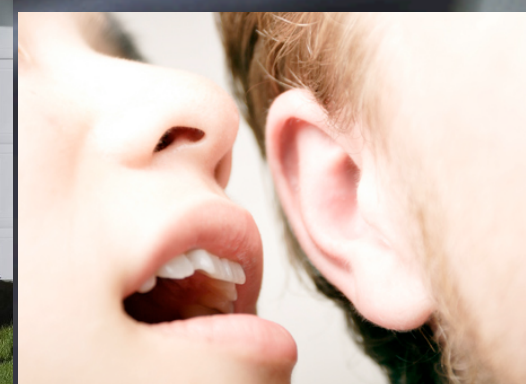
Département d'informatique et de recherche opérationnelle

Neuro-cognitive inspiration

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







Local vs Distributed Representation

Debate since early 80's
(connectionist models)

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

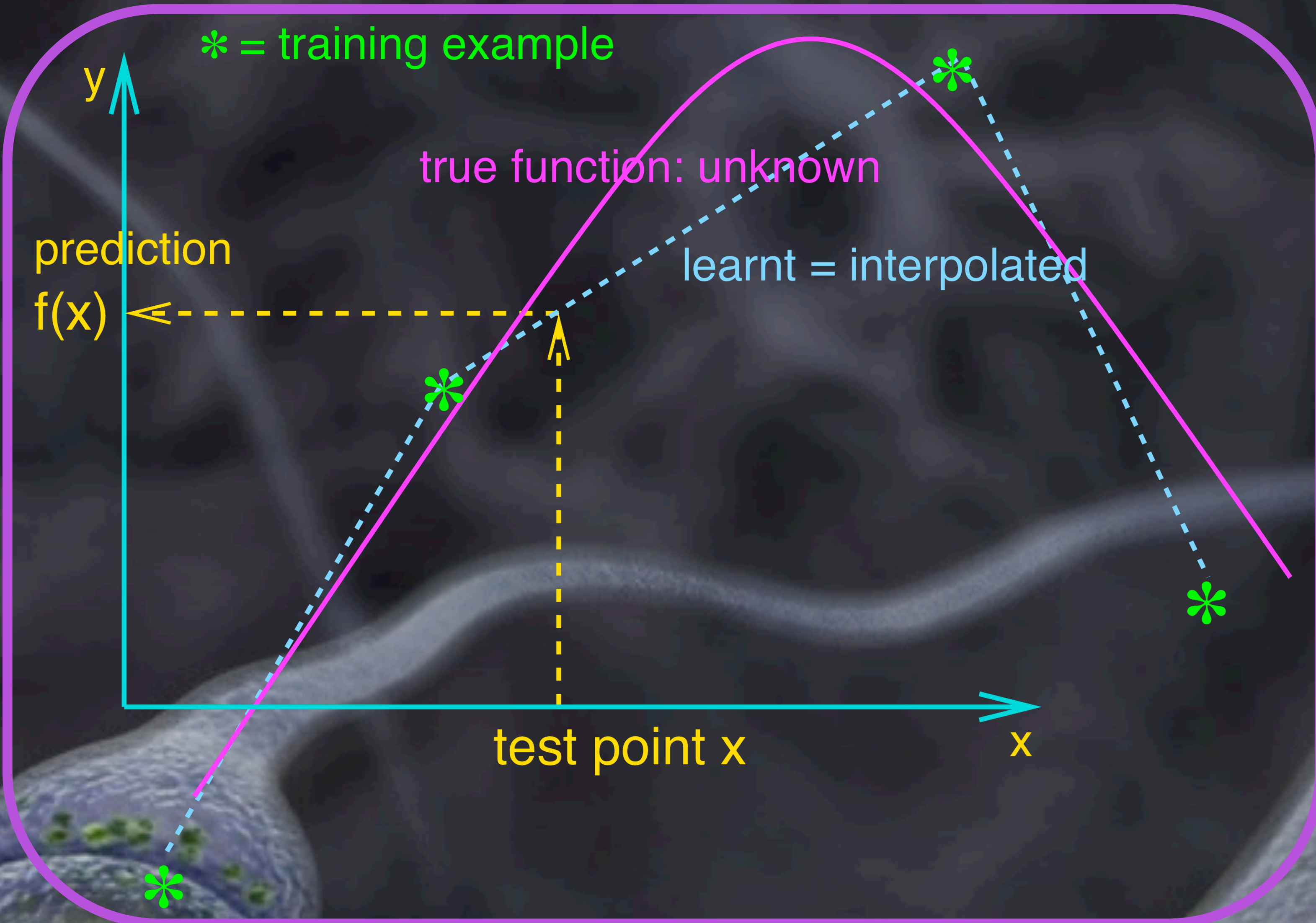
Distributed representations:
- ≈ 1% active neurons in brains
- exponentially more efficient
- difficult optimization

What is Learning?

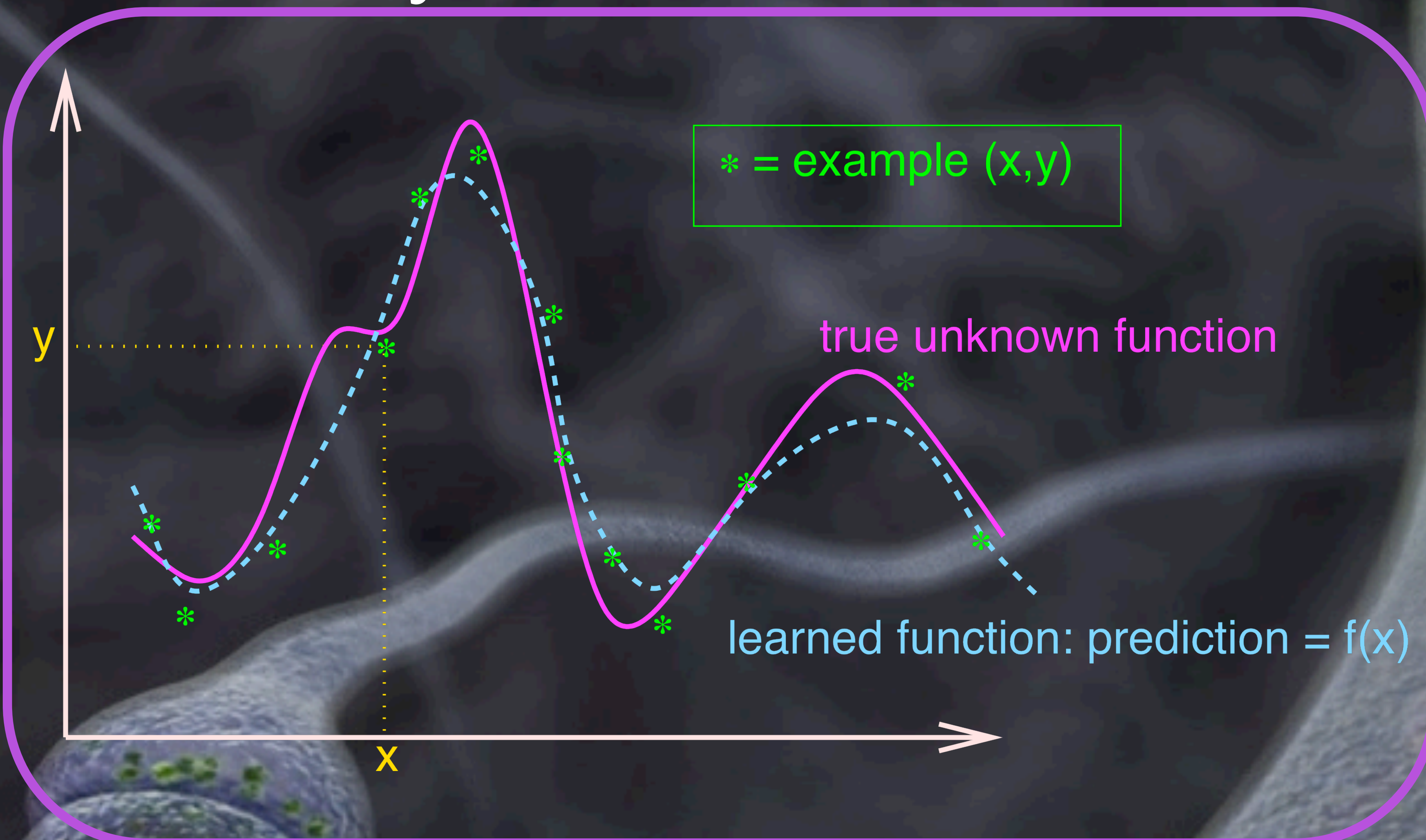
Learn underlying and previously unknown structure, from examples

= CAPTURE THE VARIATIONS

Locally capture the variations

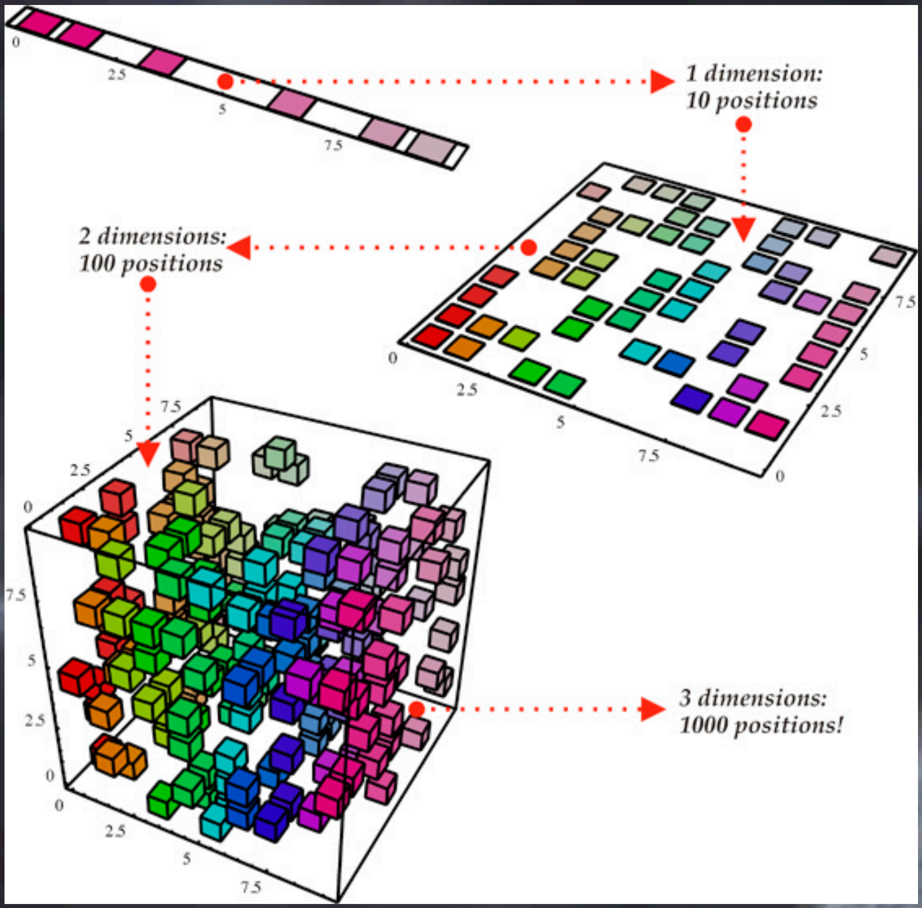


Easy when there are only a few variations

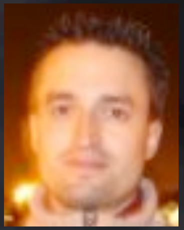


Curse of dimensionnality

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



Theoretical results



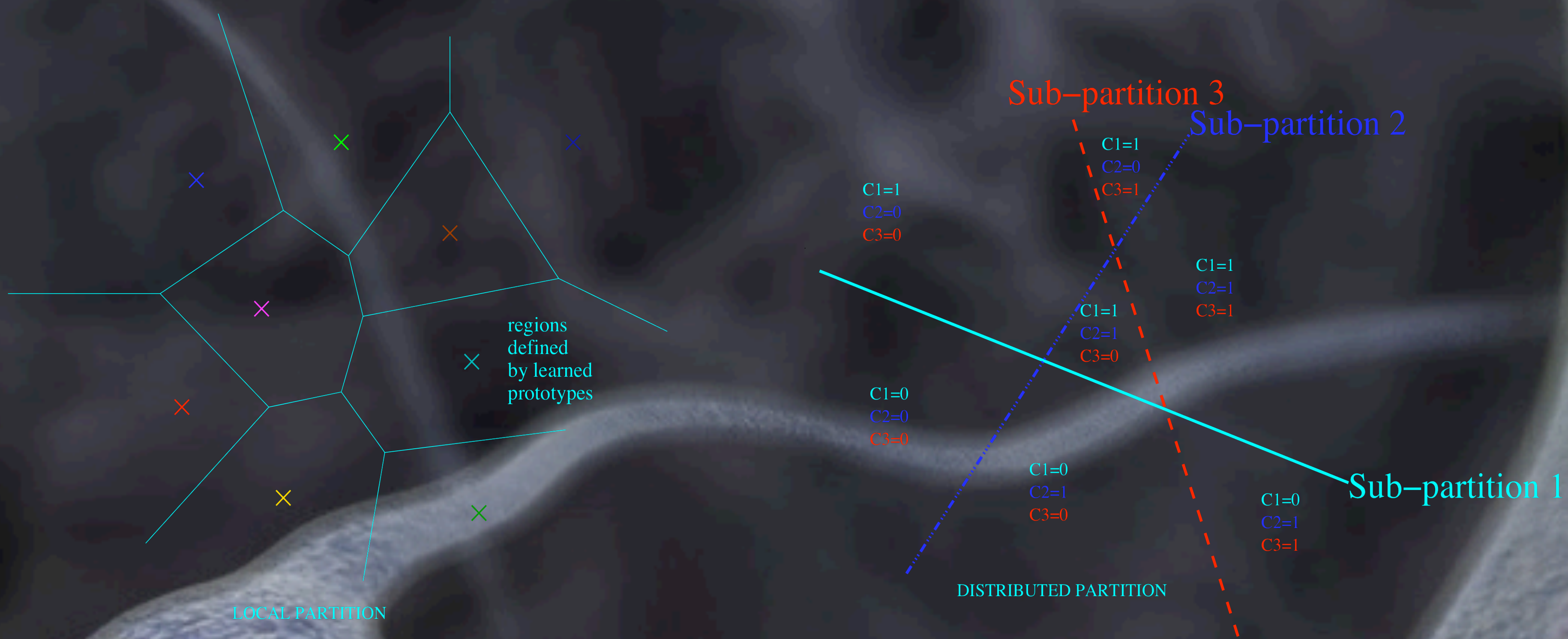
Olivier Delalleau

- **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

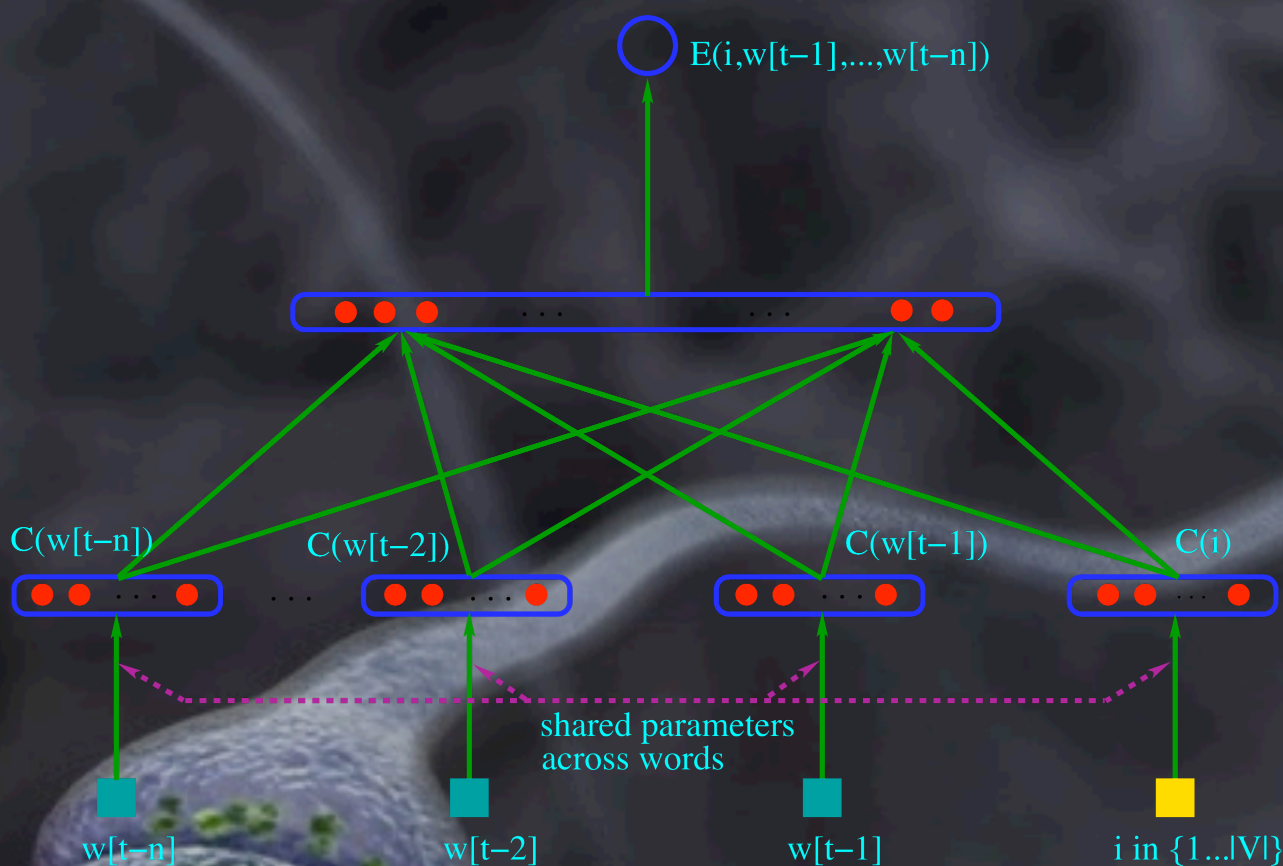


Neurally Inspired Language Models

- 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

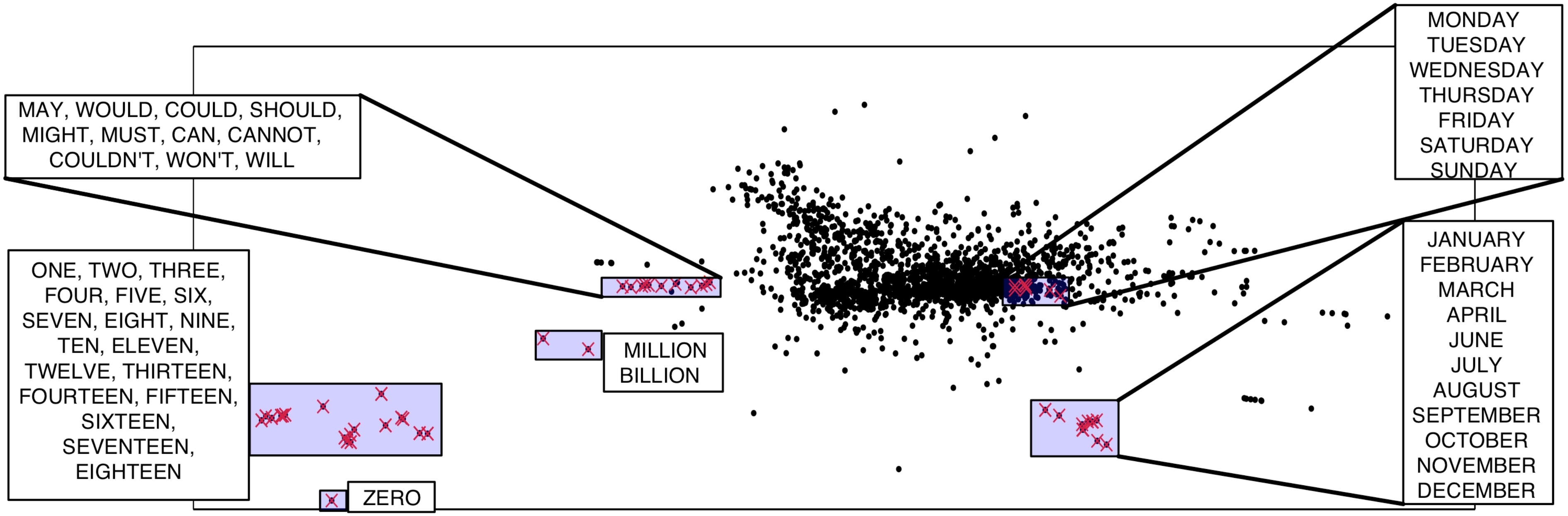
P(w[t]=i | context) = exp(−E(i,w[t−1],...,w[t−n])) / sum_j exp(−E(j,w[t−1],...,w[t−n]))
= softmax(−E(.,w[t−1],...,w[t−n]))



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

Bengio et al 2003, Schwenk et al 2005, Collobert & Weston, ICML'08

Embedding Symbols

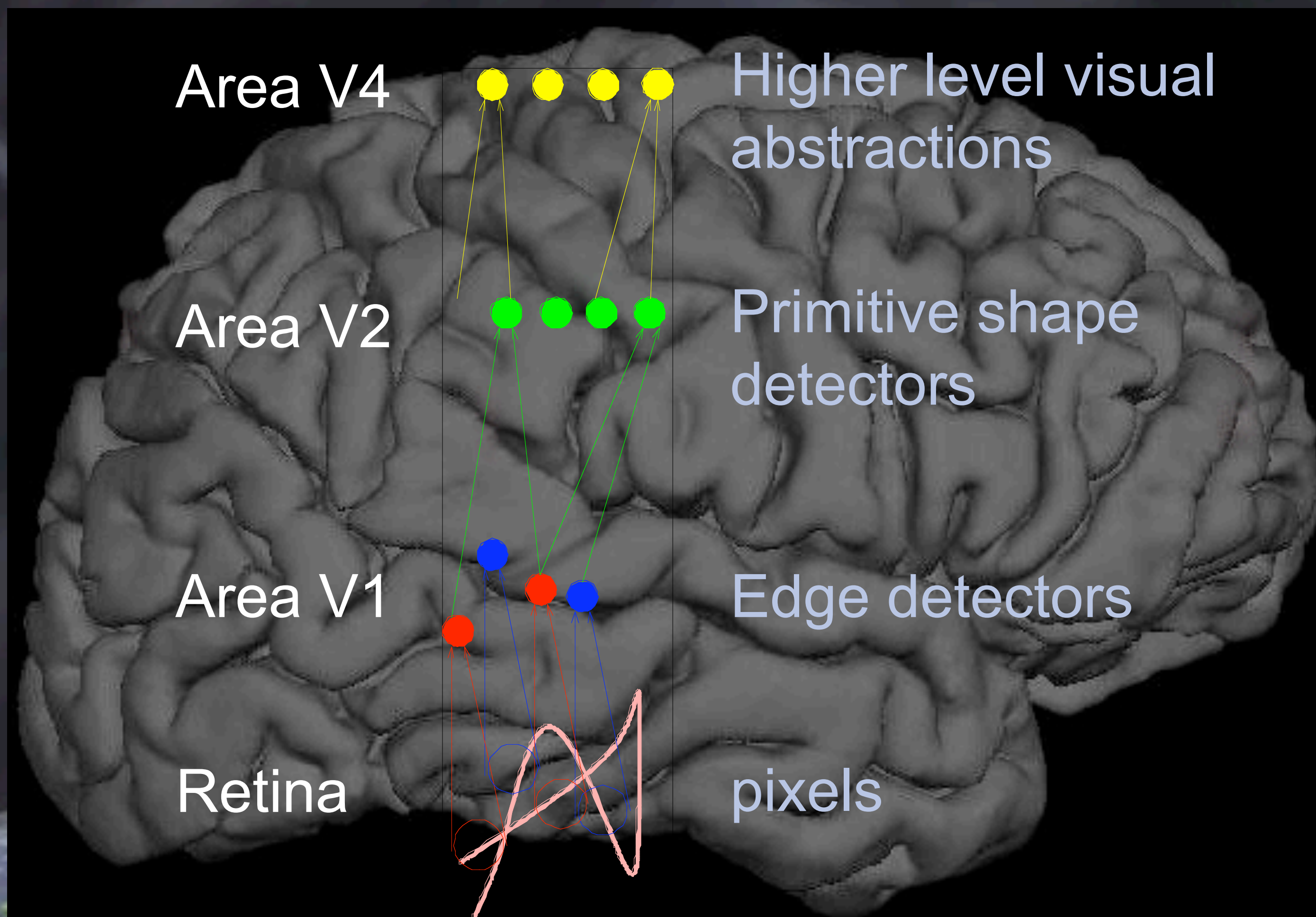


Blitzer et al 2005, NIPS

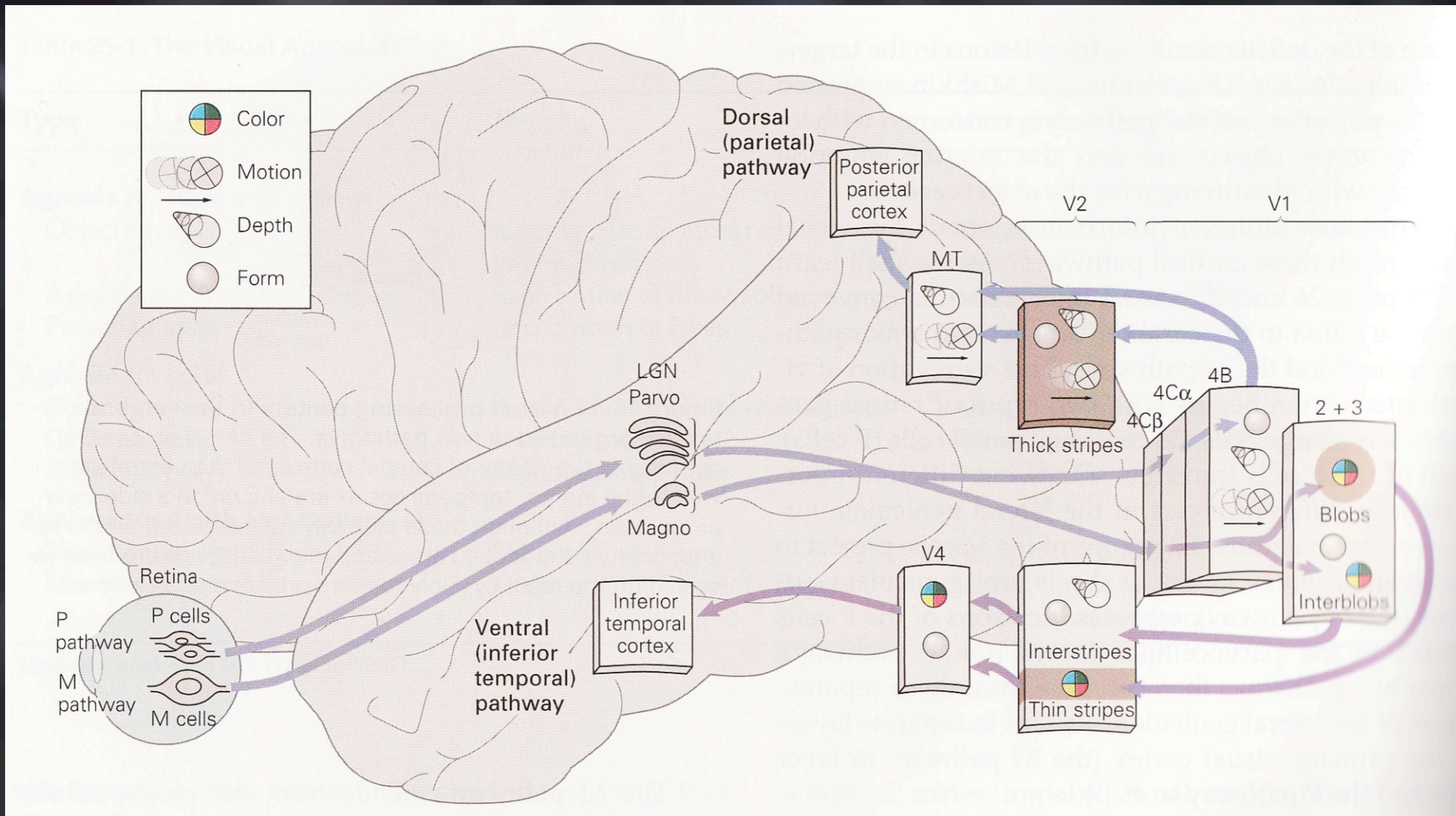
Nearby Words in Semantic Space

Show t-SNE embeddings of *Collobert & Weston (ICML'08)*, done by Joseph Turian

Deep Architecture in the Brain



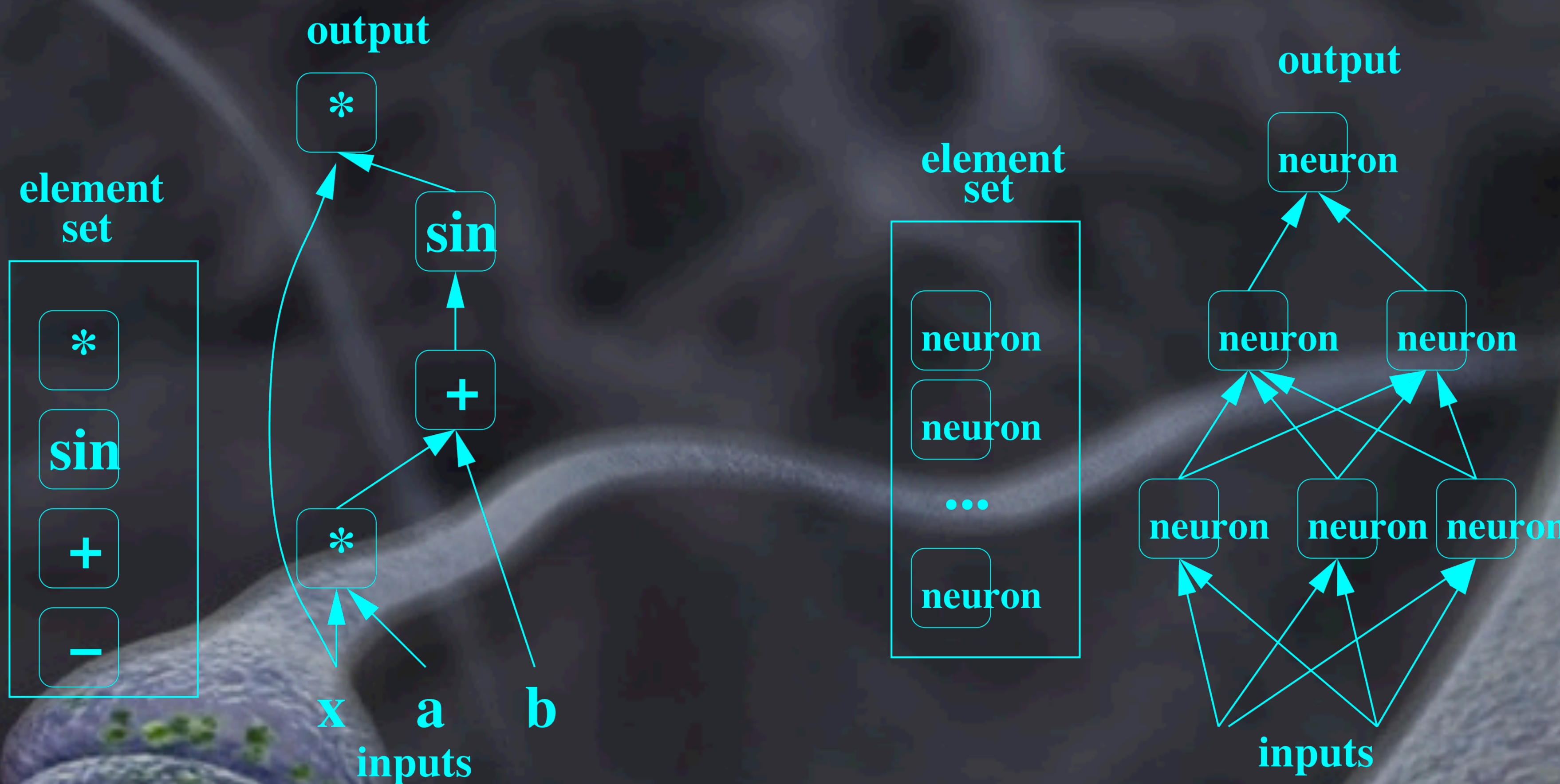
Visual System



Sequence of transformations / abstraction levels

Architecture Depth

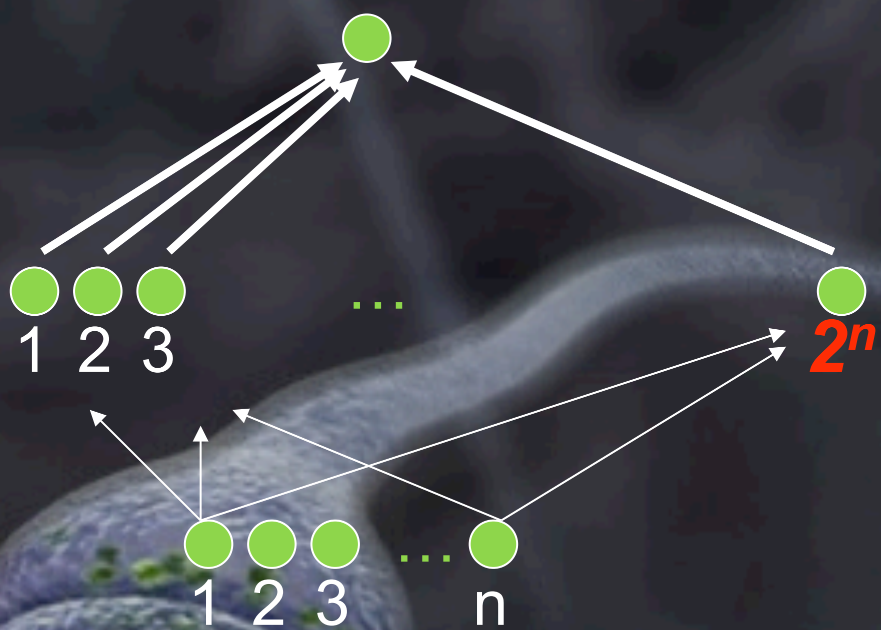
Computation performed by learned function can be decomposed into a graph of simpler operations



Insufficient Depth

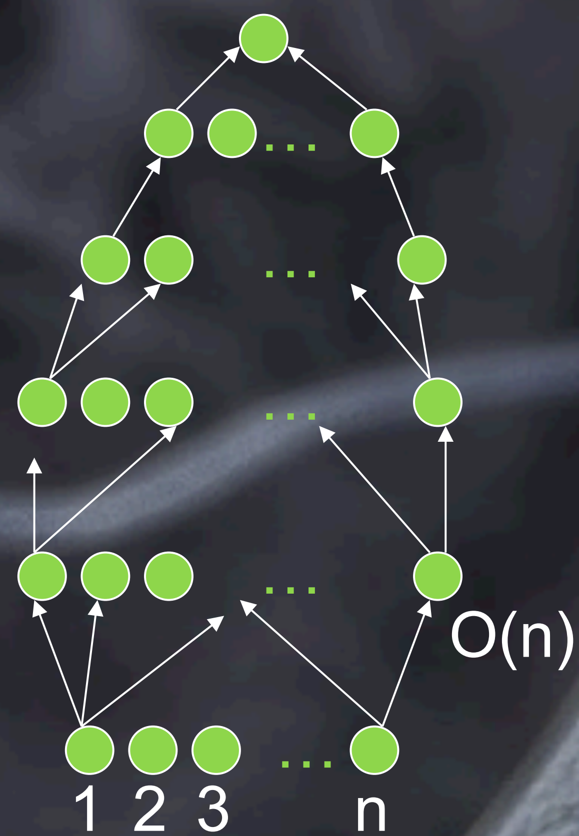
Insufficient depth =

May require exponential-size architecture



Sufficient depth =

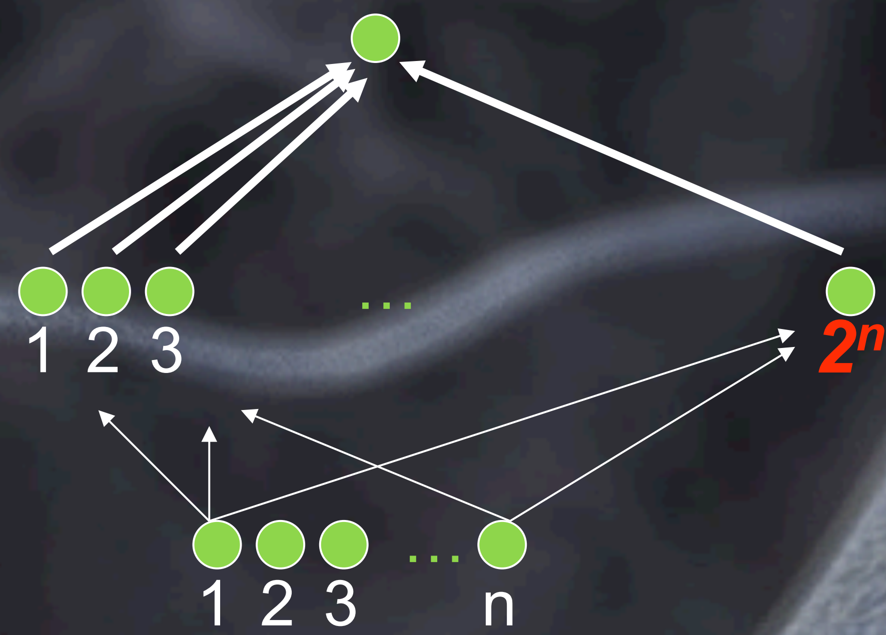
Compact representation



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



Breakthrough!

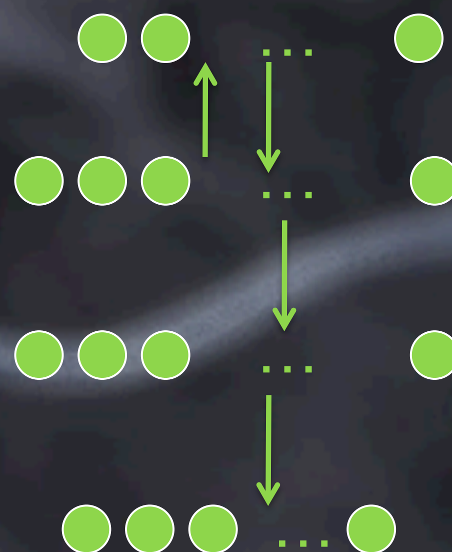
Before 2006

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)

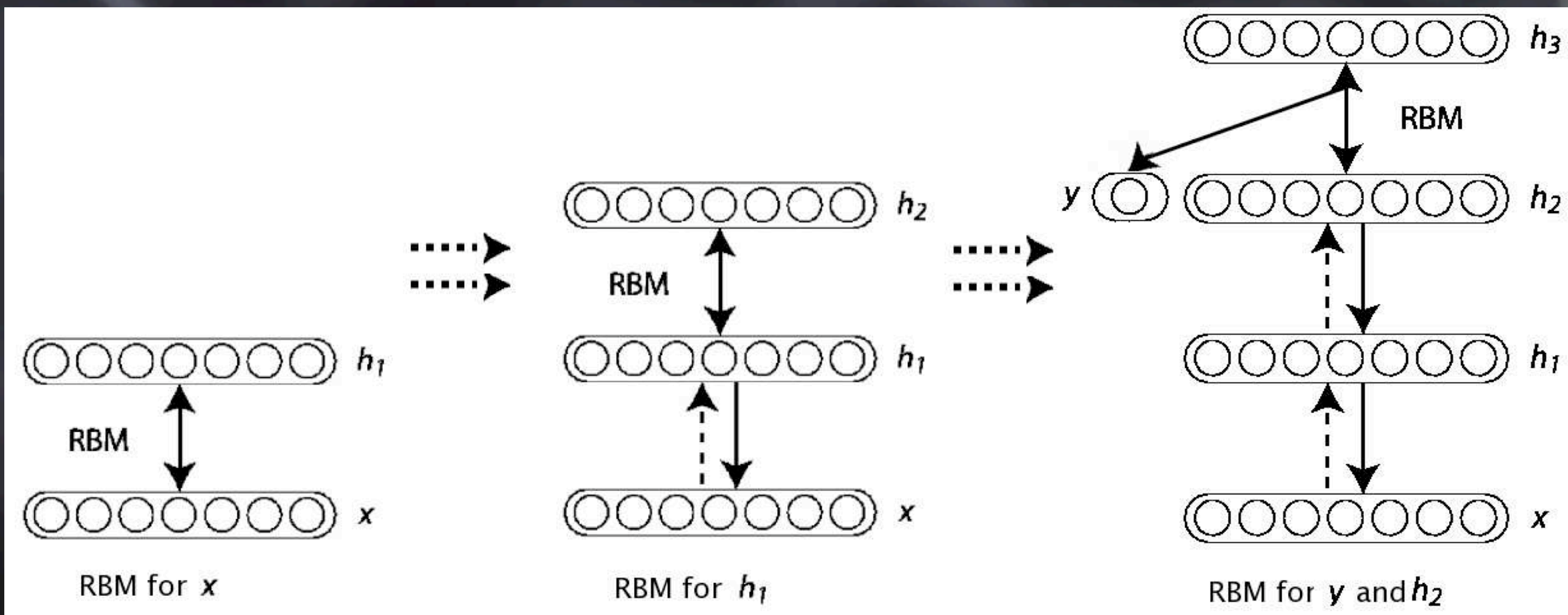


Success of deep distributed neural networks

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)

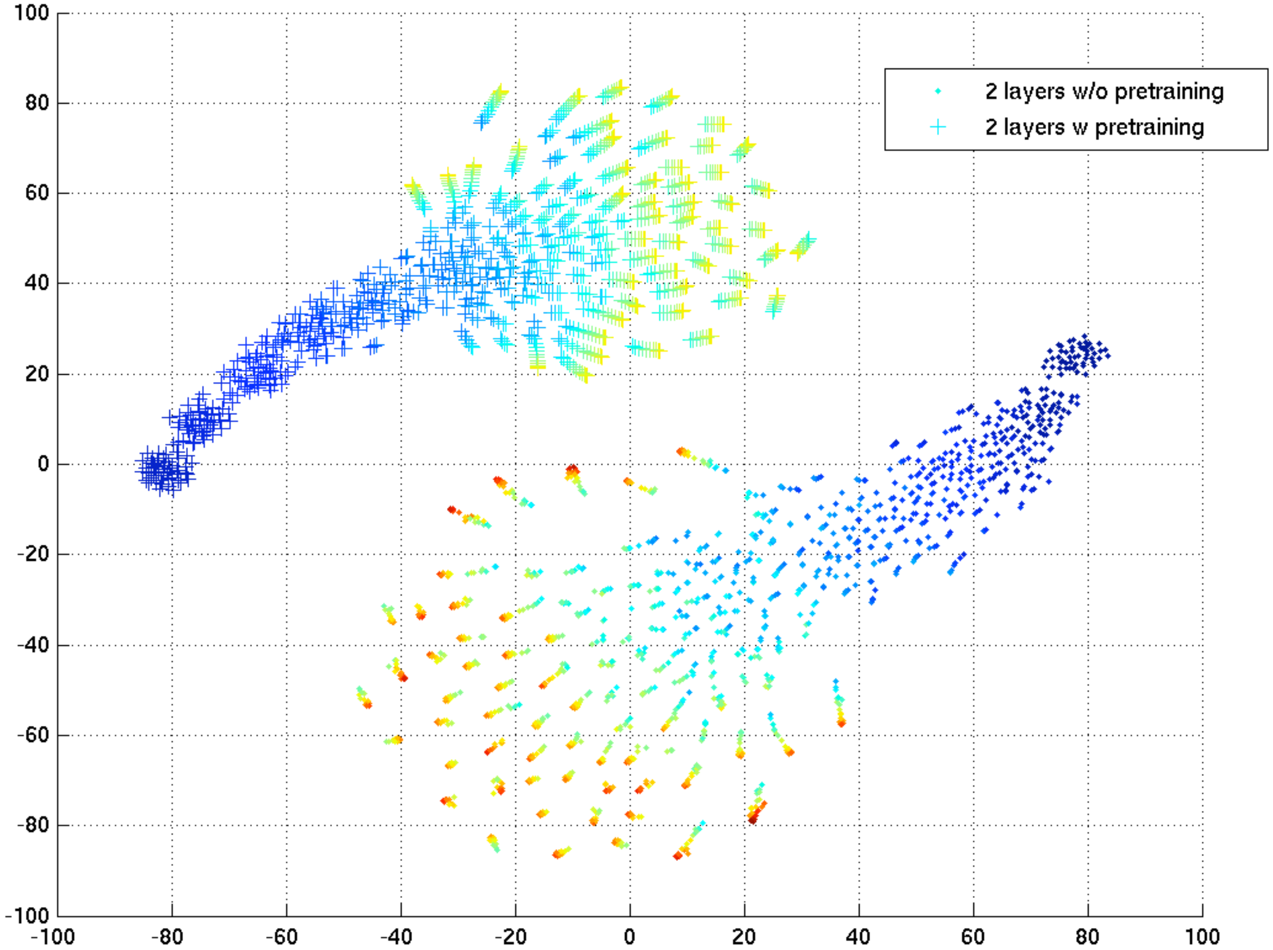
Unsupervised greedy layer-wise pre-training



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
- pre-training moves into improbable region with better basins of attraction
- Training one layer after the other ≈ continuation method (Bengio 2008)

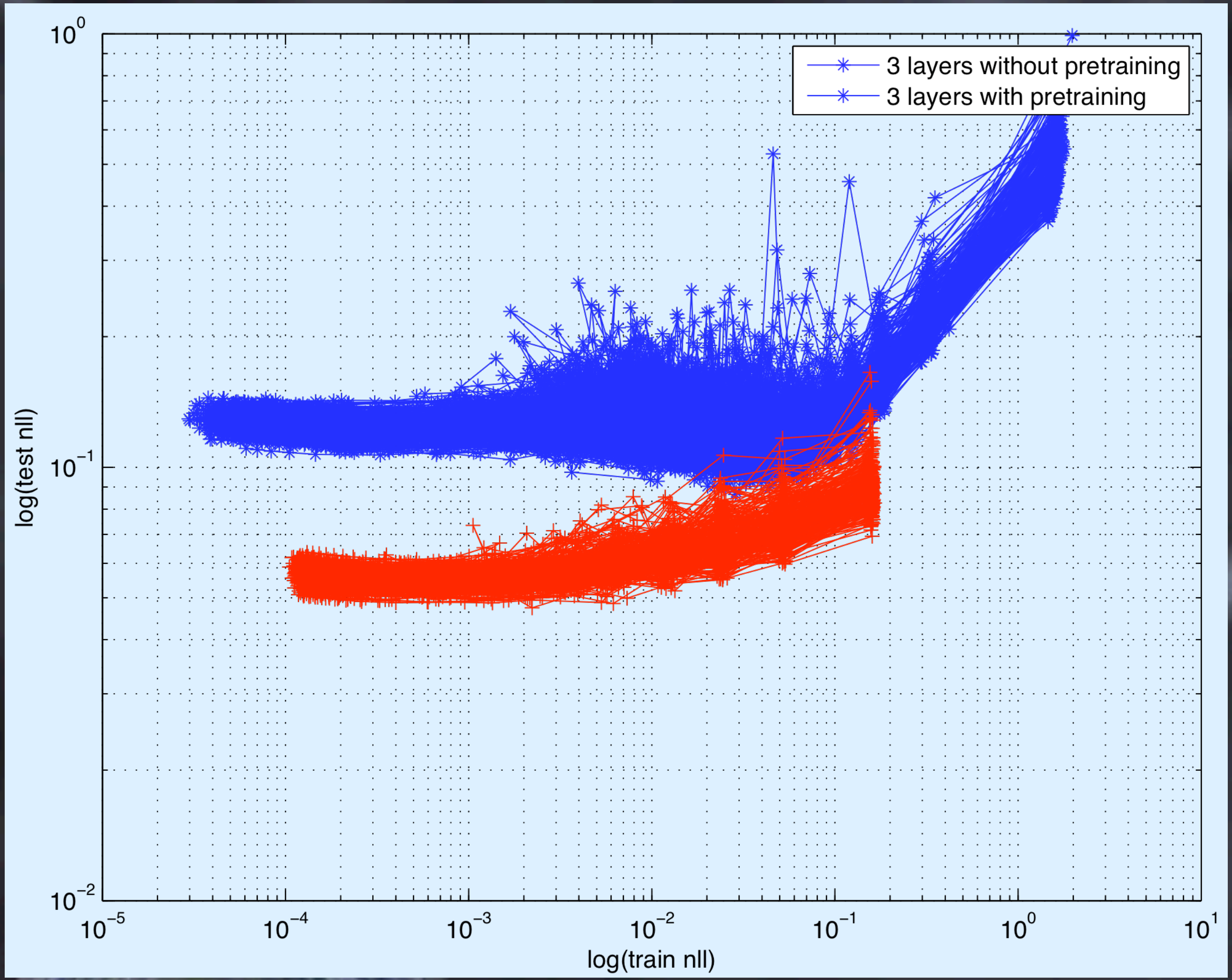
Flower Power



Dumitru Erhan

Pierre-Antoine Manzagol

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.

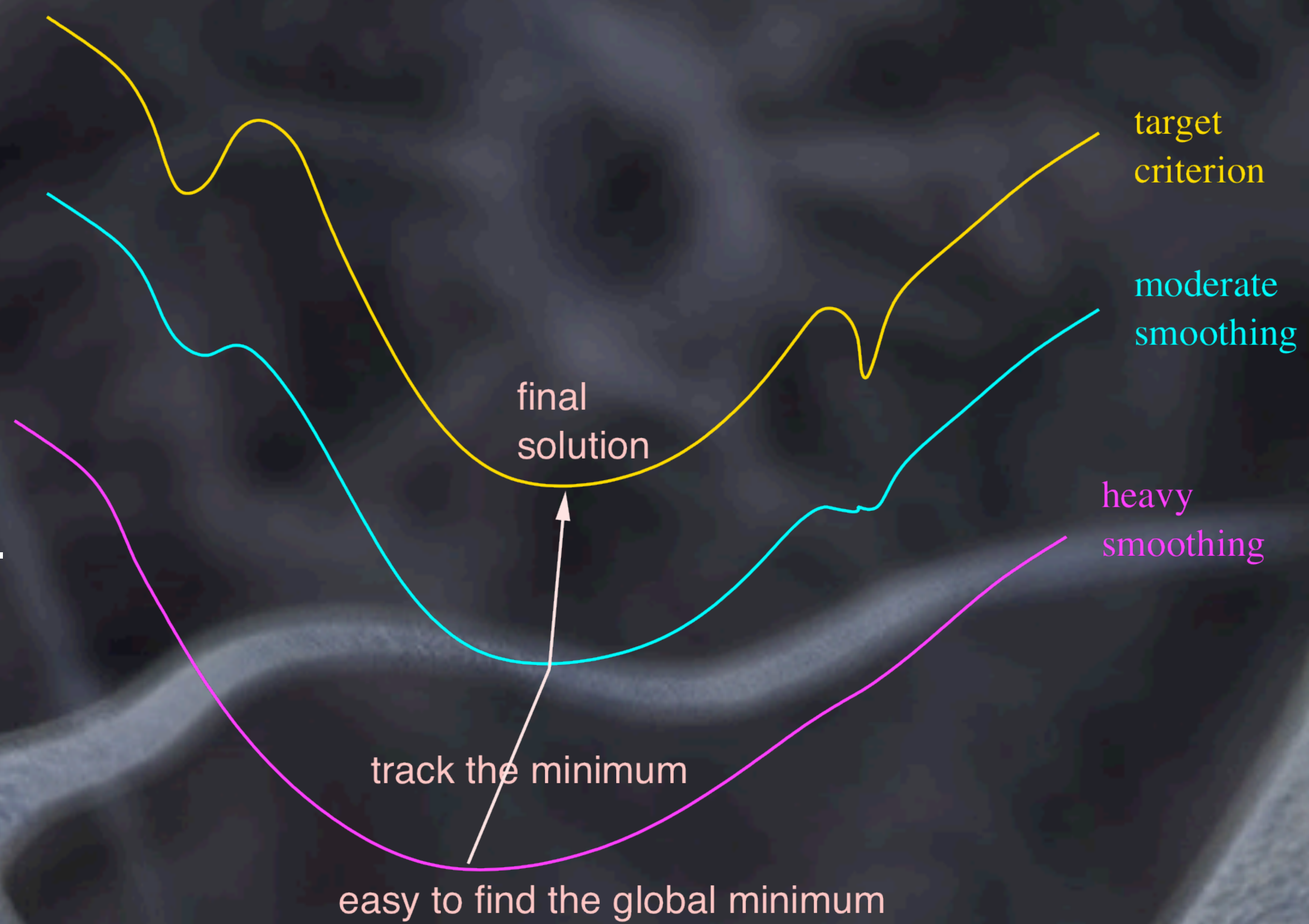
Non-convex optimization

- 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

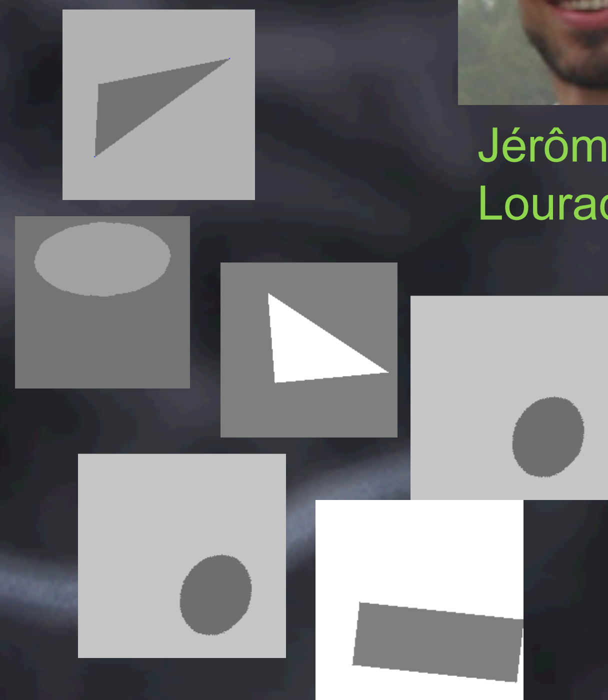
First learn simpler tasks, then build on top and learn higher-level abstractions.

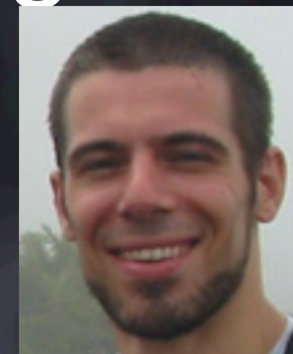


Experiments on multi-stage curriculum training

Stage 1 data:

Stage 2: data

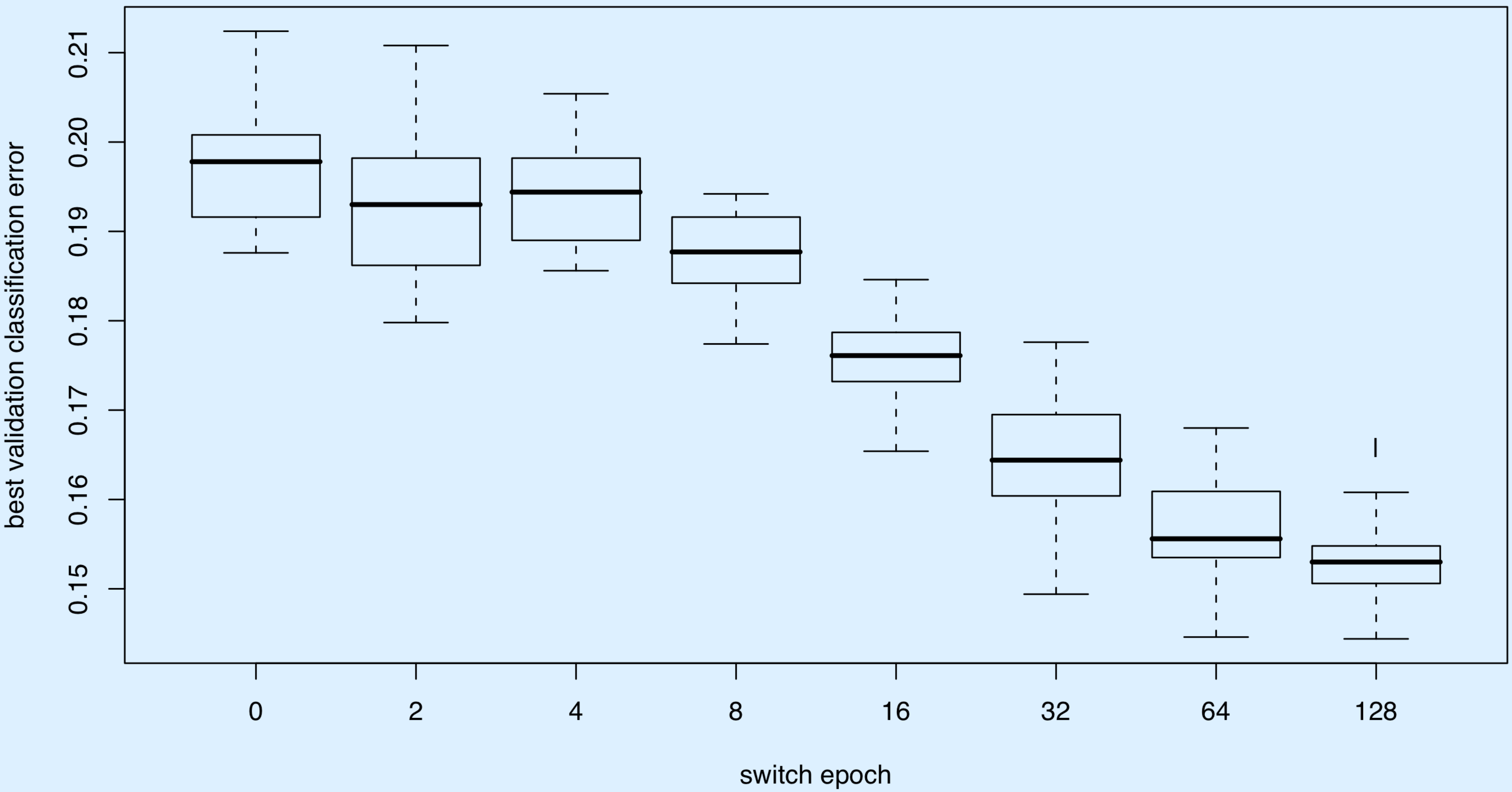




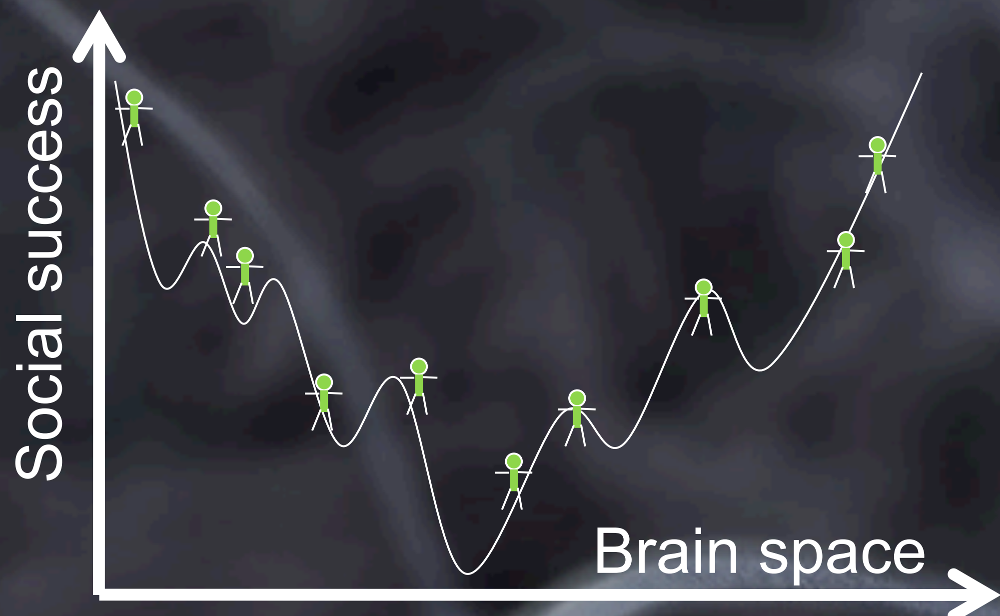
Jérôme Louradour

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



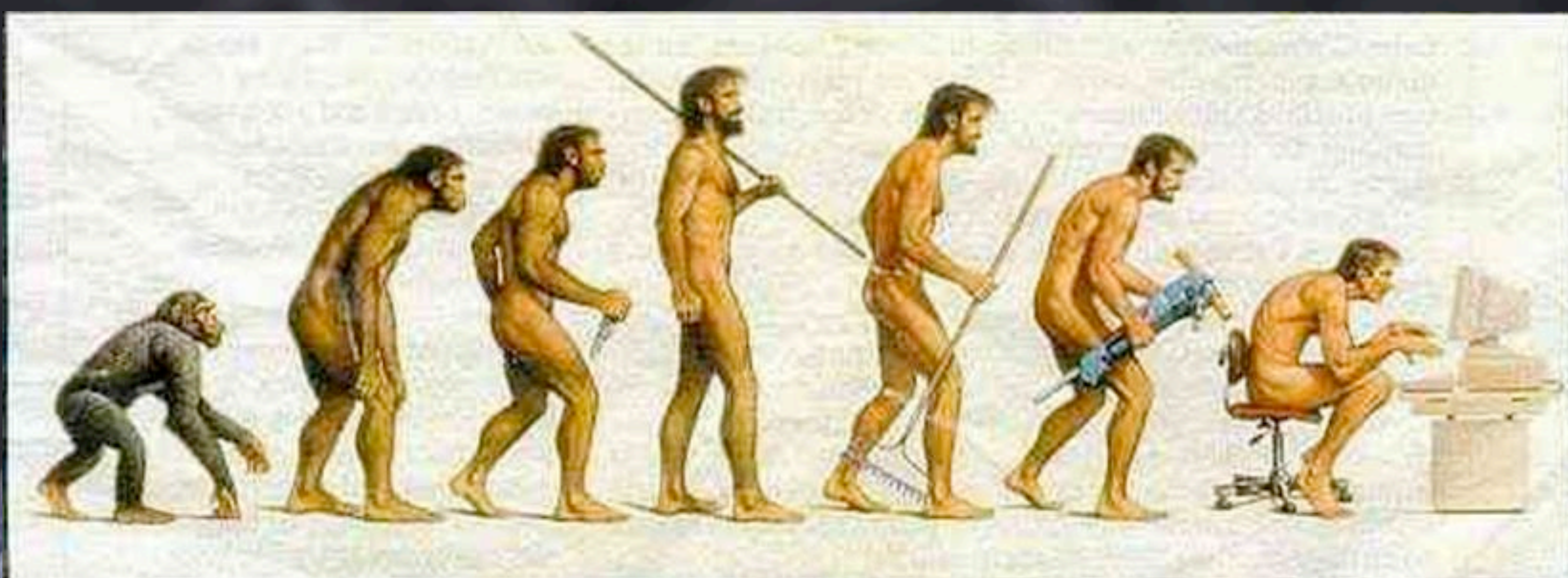
Parallelized exploration: Evolution of concepts



- 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