Learning Deep Architectures: a Stochastic Optimization Challenge?

Yoshua Bengio, U. Montreal

CIFAR NCAP Meeting, Millcroft Inn, May 9th, 2009

Discussion subjects

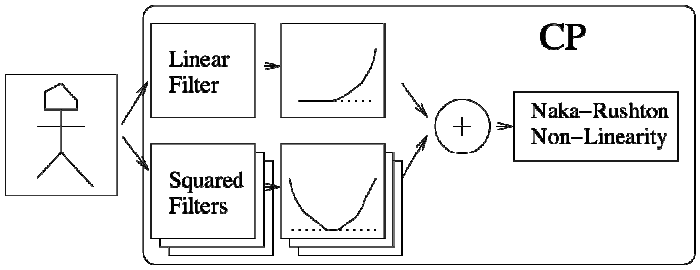
- What neuron models best for learning?
- Why is unsupervised pre-training so successful?
- What tricks to handle zillions of local minima?
- What unsupervised training principles?

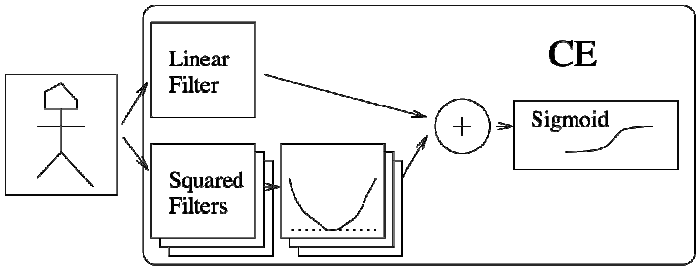
What neuron models for learning deep architectures?

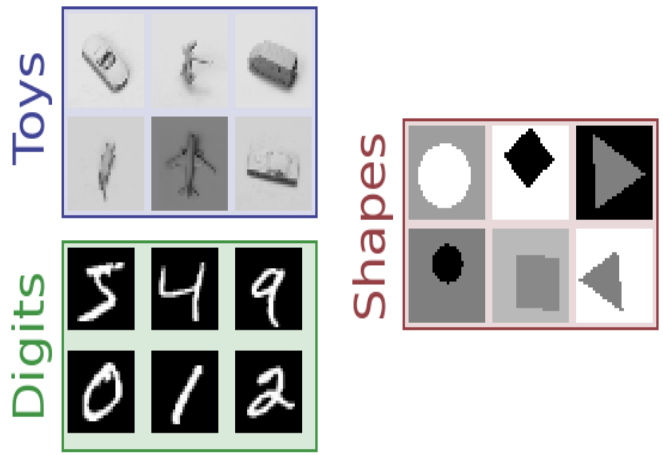
What neuron model?

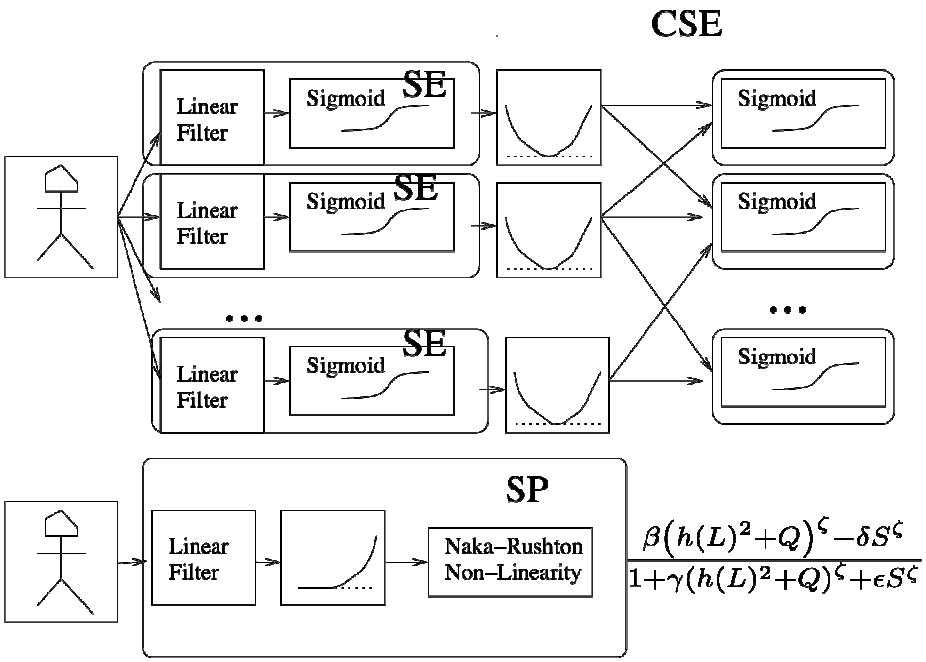
- Amount of noise / randomness in individual neuron behavior?
- Linear or higher-order computations in the dendritic tree?
- Exponentially or polynomially saturating non-linearity or Geoff's Poisson rate neurons?
- Temporal constancy? multiple time scales?
- Structure and type of feedback connections?
- Slow and fast synapses?
- Back-prop through fast feedback connections?

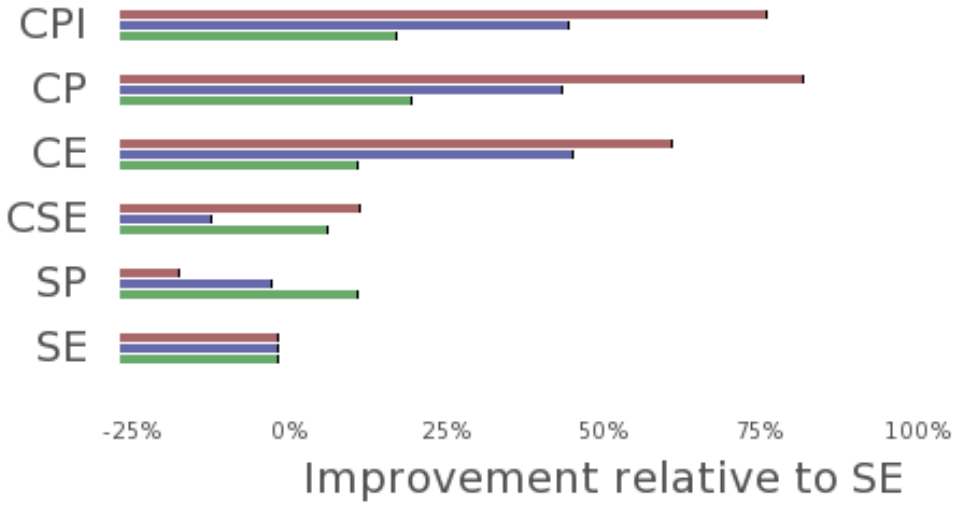
Quadratic interactions? Sigmoid?











Why is unsupervised pre-training so successful?

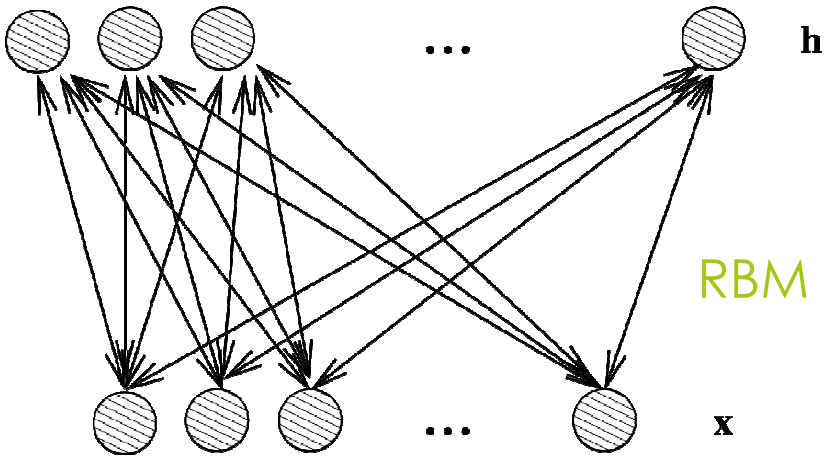
Success of deep neural networks

Since Hinton et al' 2006 DBN paper:

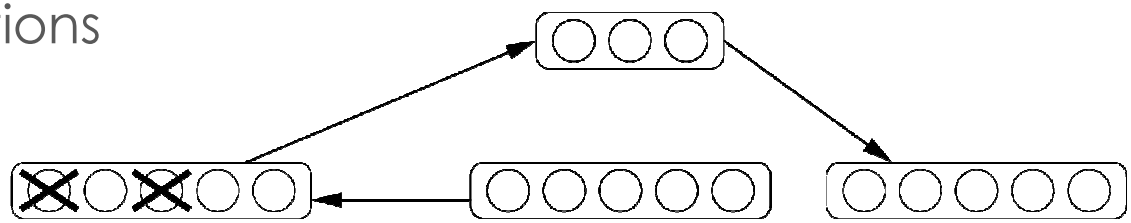
- Records broken on MNIST handwritten character recognition benchmark (Ranzato et al 2007, 2008)
- State-of-the-art beaten in language modeling (Collobert & Weston 2008)
- NSF et DARPA are interested…
- Similarities between V1 & V2 neurons and representations learned with deep nets
- Dozens of papers. See my review paper to appear in *Foundations and Trends in Machine Learning*.

RBMs and Auto-Encoders

- Building blocks of current learning algorithms for deep architectures
- Mathematically similar
- Feedback connections for learning
- Injection of noise



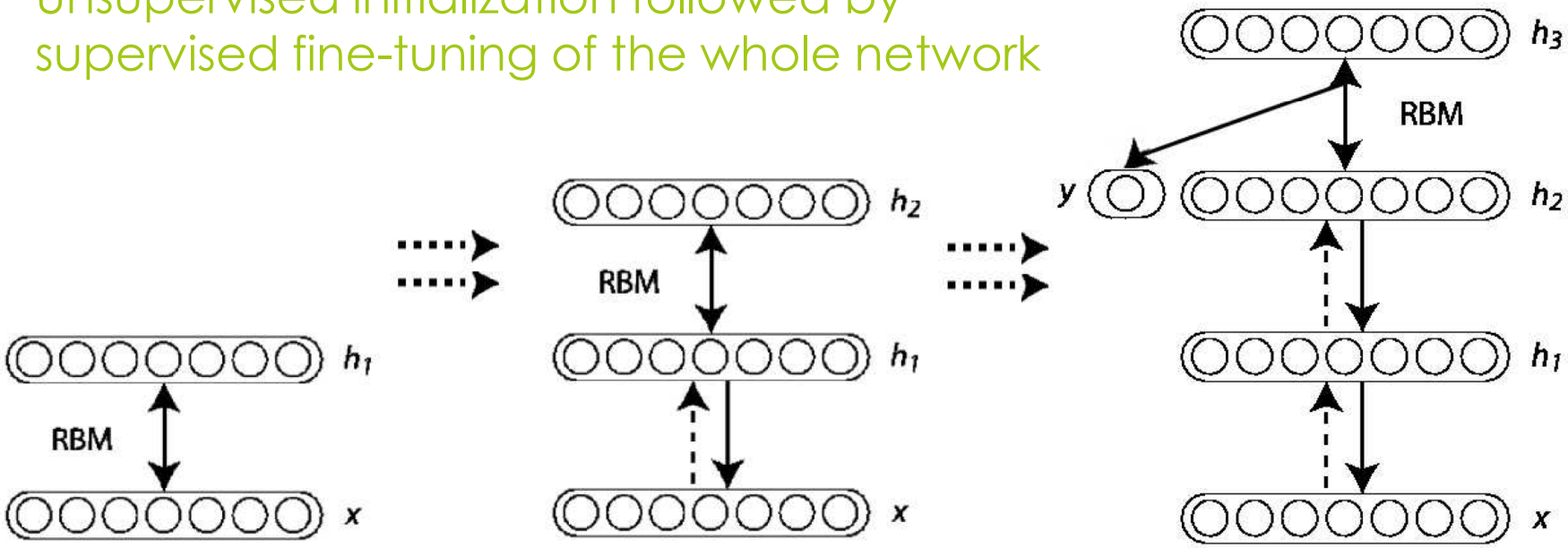
RBM



Denoising auto-encoder

Unsupervised layer-wise pre-training

Unsupervised initialization followed by
supervised fine-tuning of the whole network



Easy

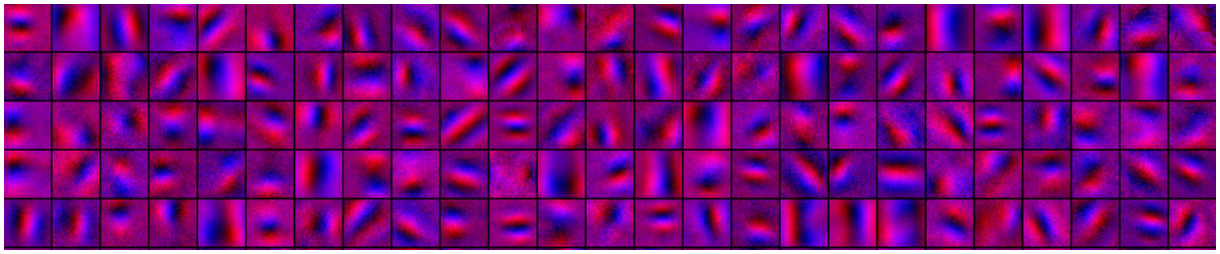
More difficult

Two phases? Pre-training + fine-tuning

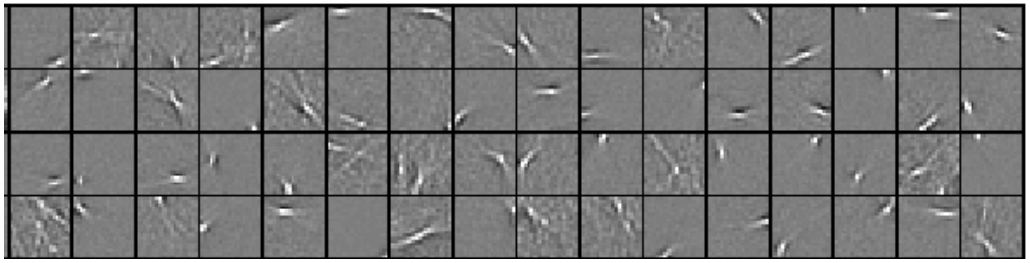
- Currently best results generally obtained when doing purely supervised fine-tuning after unsupervised pre-training
- Kind of disappointing
- Can we avoid the fine-tuning alltogether?
- Can we fold both phases together? (would be very useful for online learning on huge datasets)

V1 and V2-like filters learned

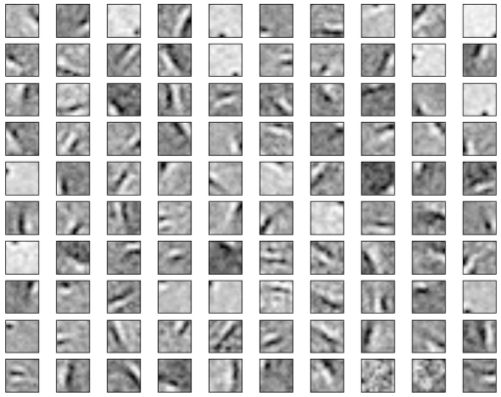
Slow features
1st layer



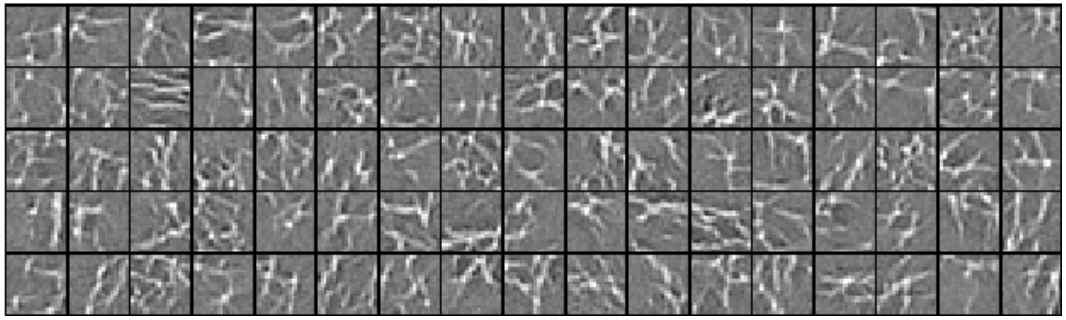
RBM 1st layer



Denoising auto-encoder
1st layer

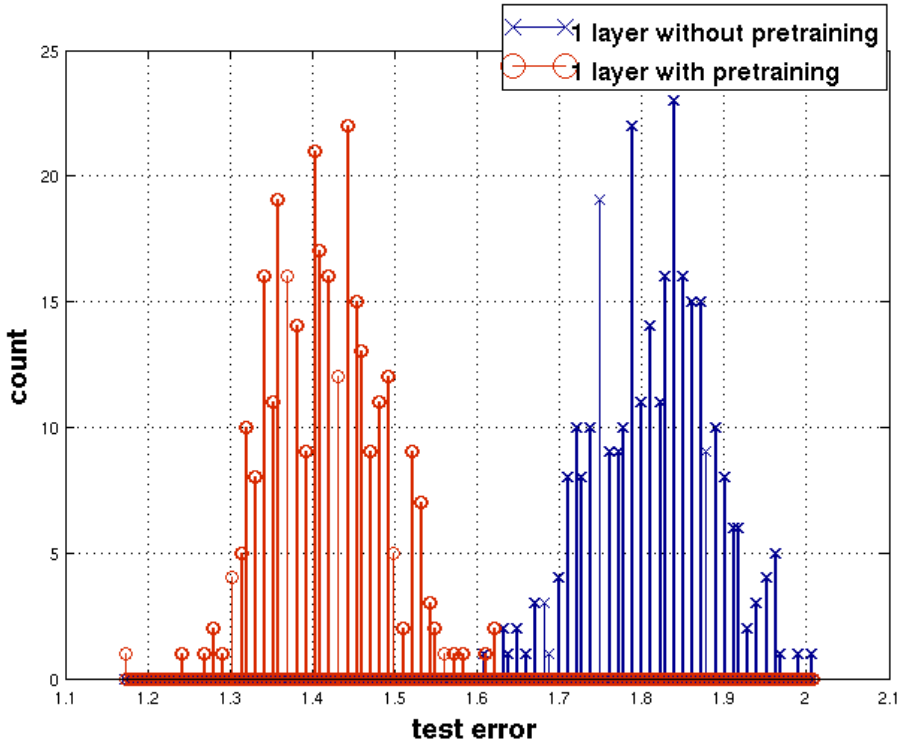


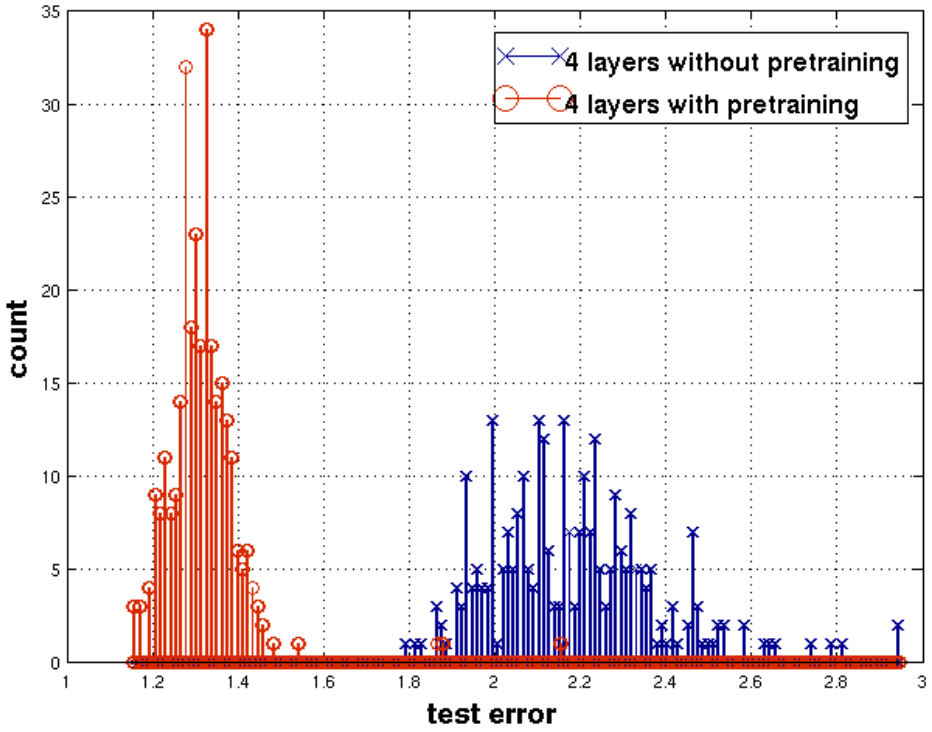
DBN
2nd
layer



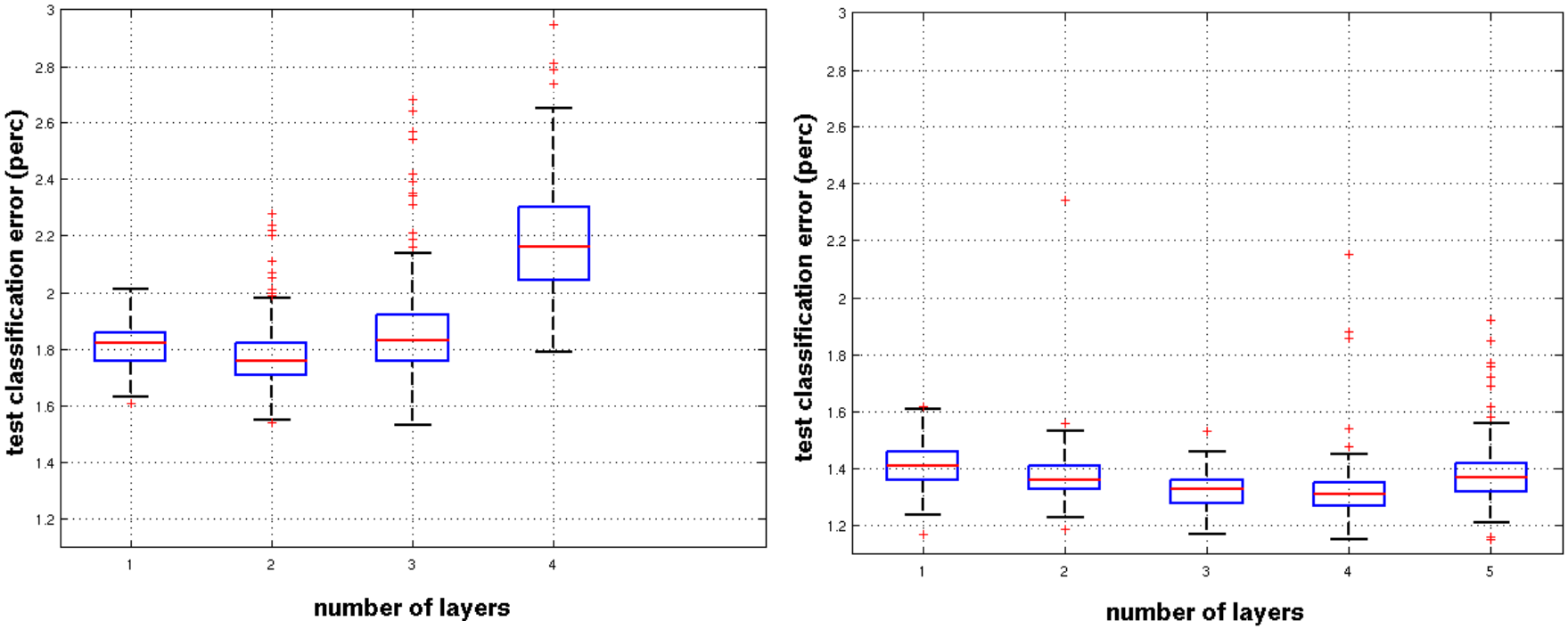
AISTATS'2009

Effect of unsupervised pre-training





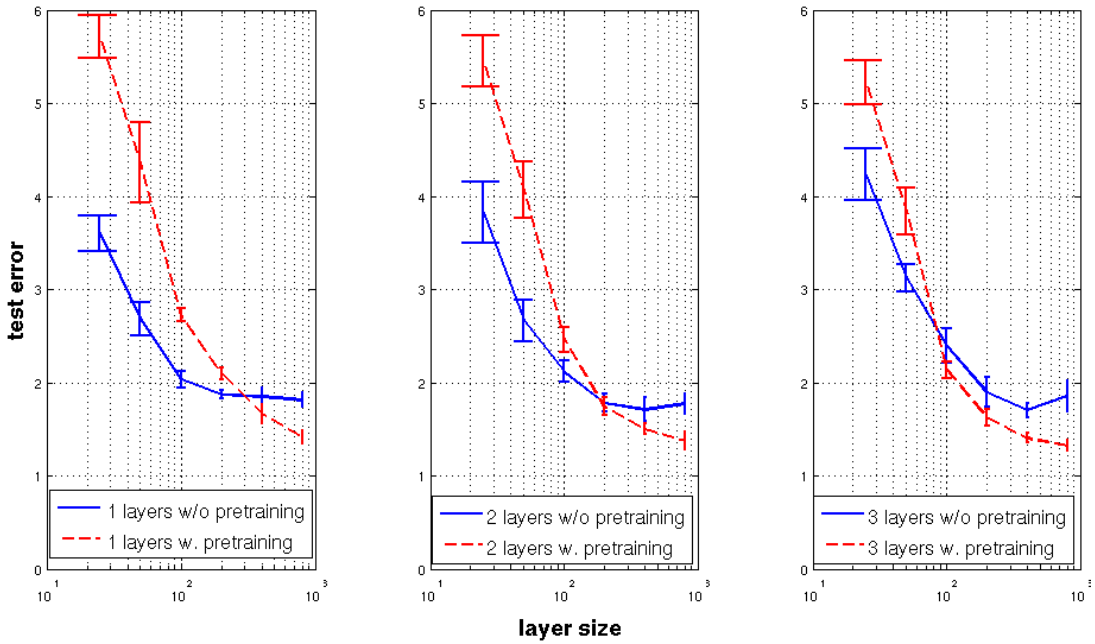
Effect of depth



Regularization or optimization?

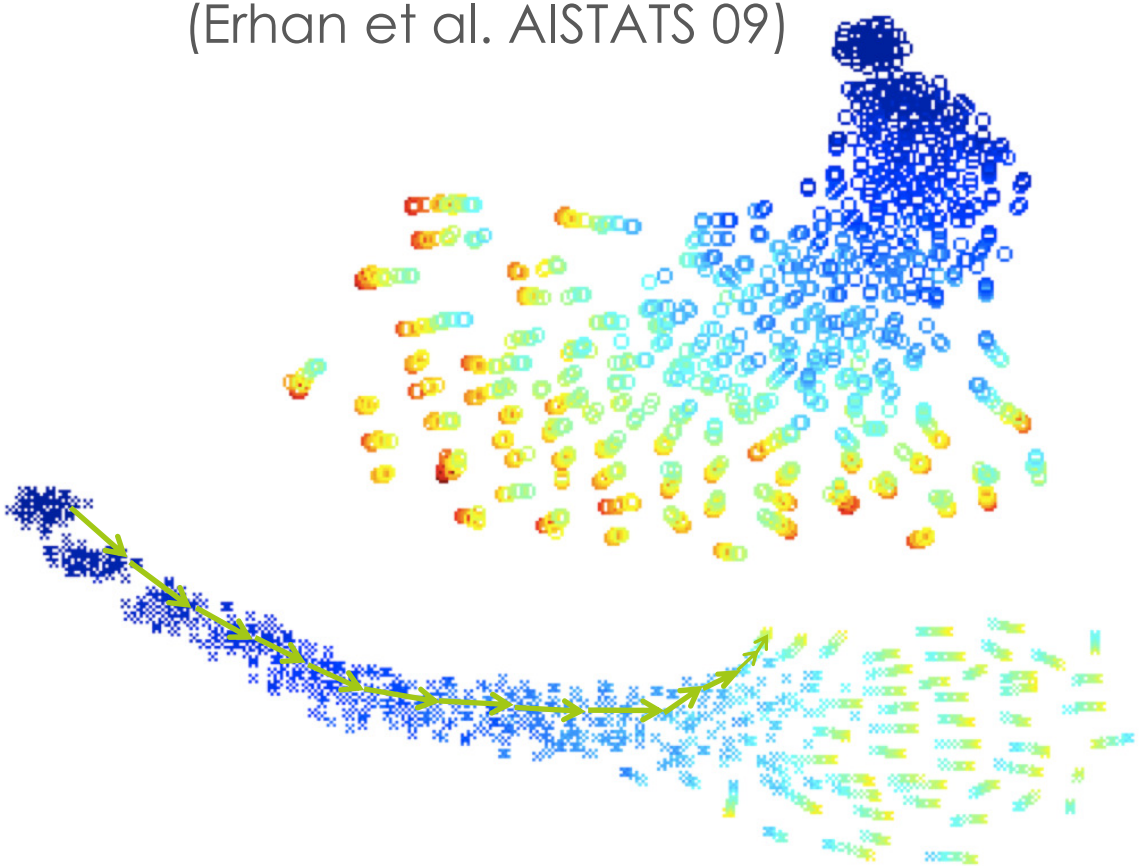
Initial results with supervised deep architectures trained with unsupervised pre-training show regularization effect:

- 0 training error with or w/o pre-training
- Pre-training hurts with too small nets



Deep training trajectories: Zillions of local minima

(Erhan et al. AISTATS 09)



Random initialization

Unsupervised guidance

$$E[\frac{\partial C(x)}{\partial \theta}] = \frac{\partial}{\partial \theta} \int C(x) p(x) dx$$

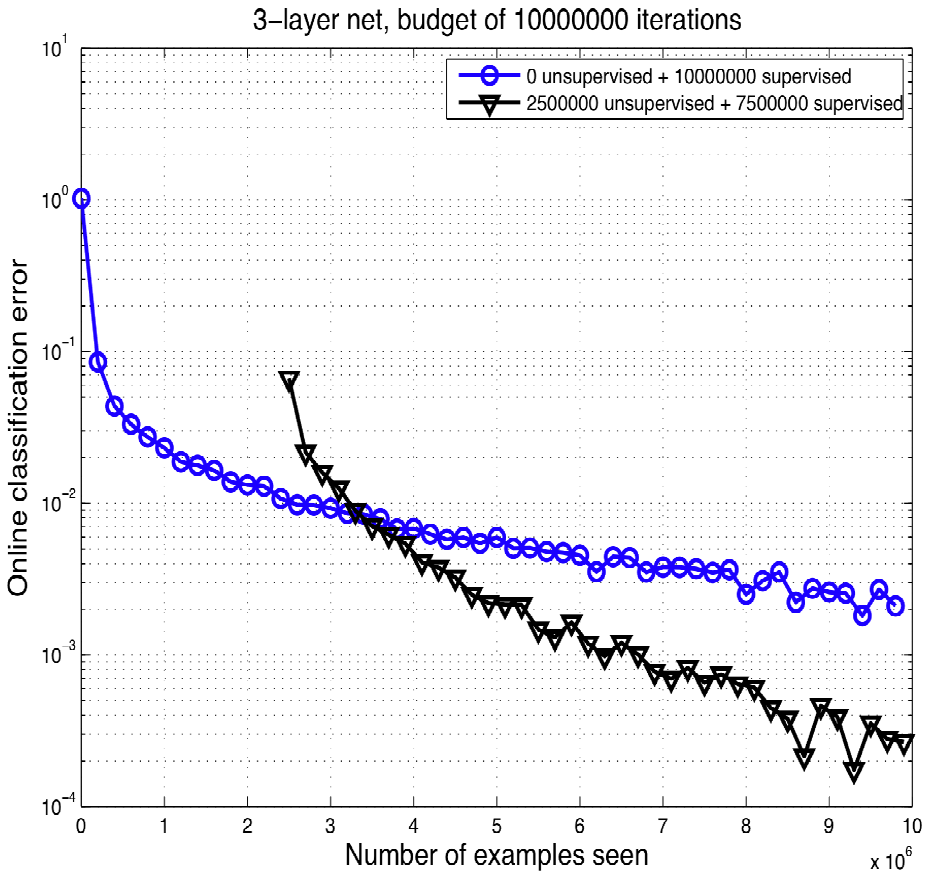
Really an Optimization Problem

- Online learning:

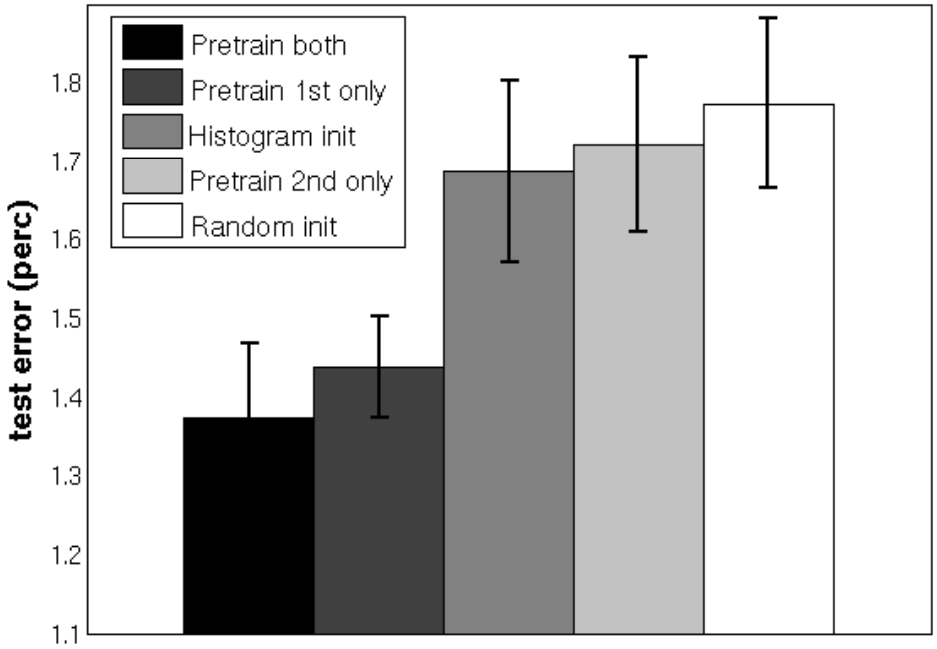
  Generalization = training objective

- If unsupervised pre-training purely a regularizer, its effect would disappear as # examples increases

- Above hypothesis contradicted by experiment



Pre-training lower layers more critical



Verifies that what matters is not just the marginal distribution over initial weight values (Histogram init.)

Why is unsupervised pre-training working?

- Regularizer or better optimization? both
- Learning mostly layer-local with unsupervised learning: hints to hidden layers
- Deep better than shallow when many factors of variation (Larochelle et al ICML'2007)
- Finds local minima that give better generalization
- Moves into improbable region with better basins of attraction, adds prior on P(input)

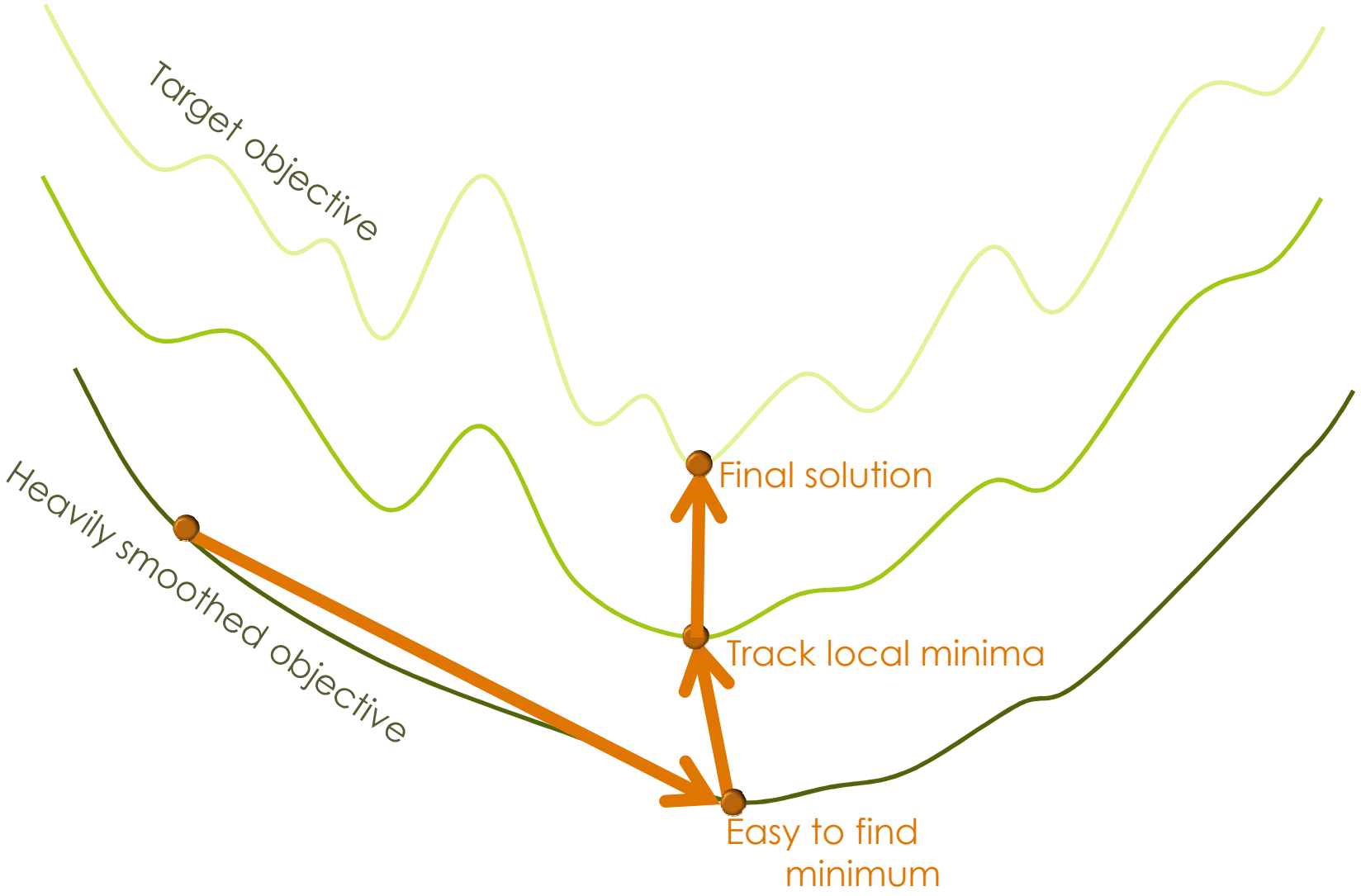
What optimization tricks?

- Humans somehow find a good solution to an intractable non-convex optimization problem.

  How?

- Guiding the optimization near good solutions
- Guiding / giving hints to intermediate layers

Continuation Methods



The Credit Assignment Problem

- Even with the correct gradient, lower layers (far from the prediction, close to input) are the most difficult to train
- Lower layers benefit most from unsupervised pre-training
  - Local unsupervised signal = extract / disentangle factors
  - Temporal constancy
  - Mutual information between multiple modalities
- Credit assignment / error information not flowing easily?
- Related to difficulty of credit assignment through time?

Guiding the Stochastic Optimization of Representations

- Train lower levels first (DBNs)
- Start with more noise / larger learning rate (babies vs adults)
- Slow features / multiple time scales
- Cross-modal mutual information
- Curriculum / shaping
- Parallel search / culture, education & research

Curriculum Learning

Guided learning helps training humans and animals

Start from simpler examples / easier tasks (Piaget 1952, Skinner 1958)

ICML'2009

Curriculum Learning

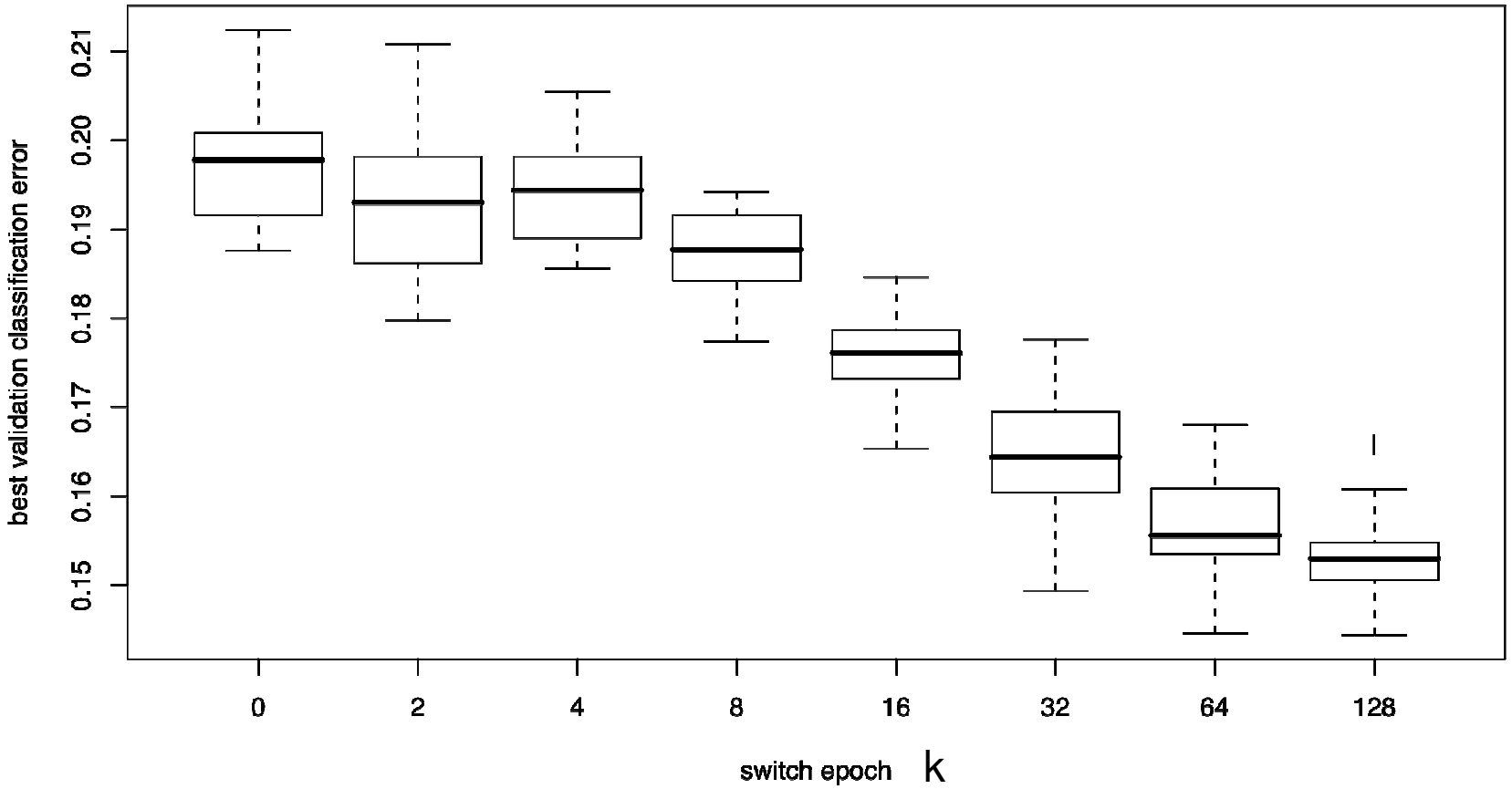
- Sequence of training distributions
- Initially peaking on easier / simpler ones
- Gradually give more weight to more difficult ones until reach target distribution

Shape Recognition

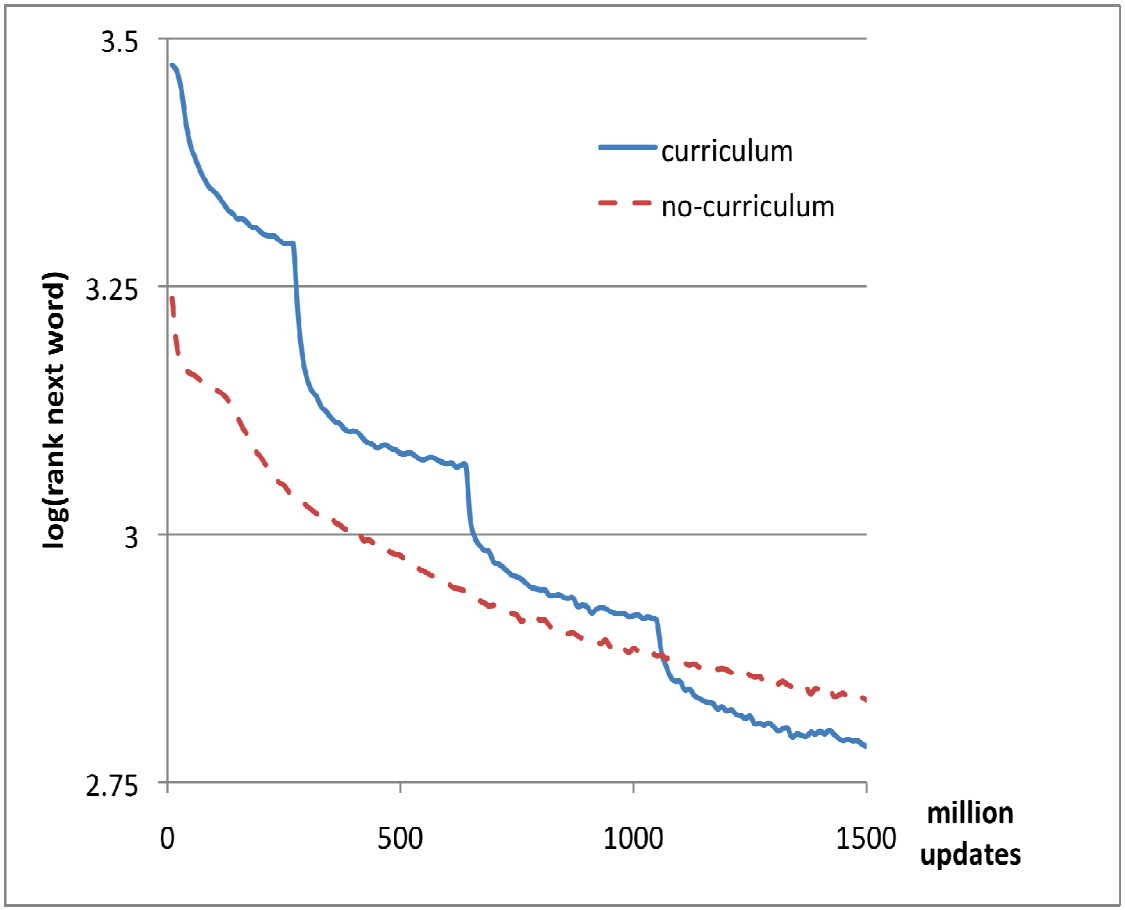
First: easier, basic shapes

Second = target: more varied geometric shapes

Shape Recognition Results

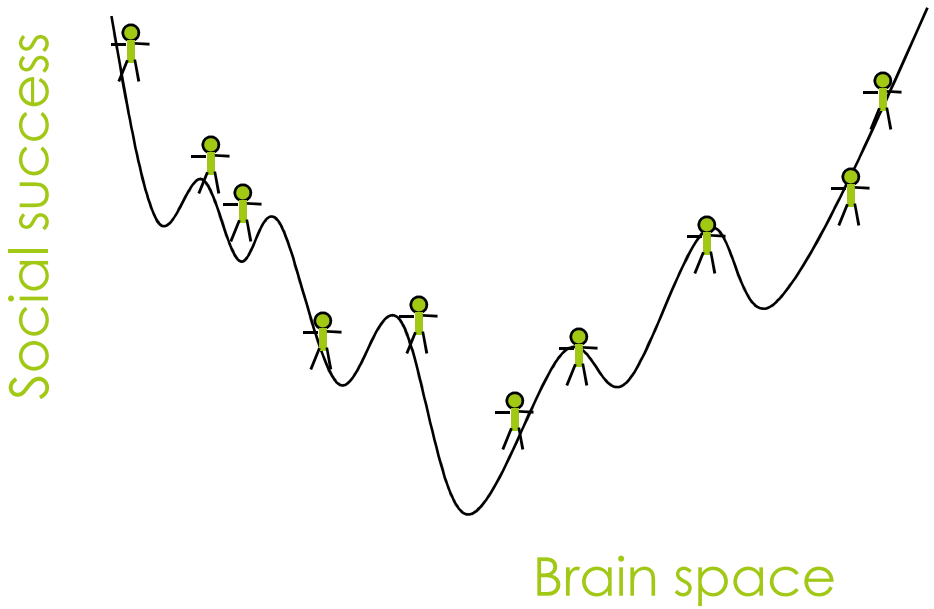


Language Modeling Results



- Gradually increase the vocabulary size (dips)
- Train on Wikipedia with sentences containing only words in vocabulary

Parallelized exploration in brain space

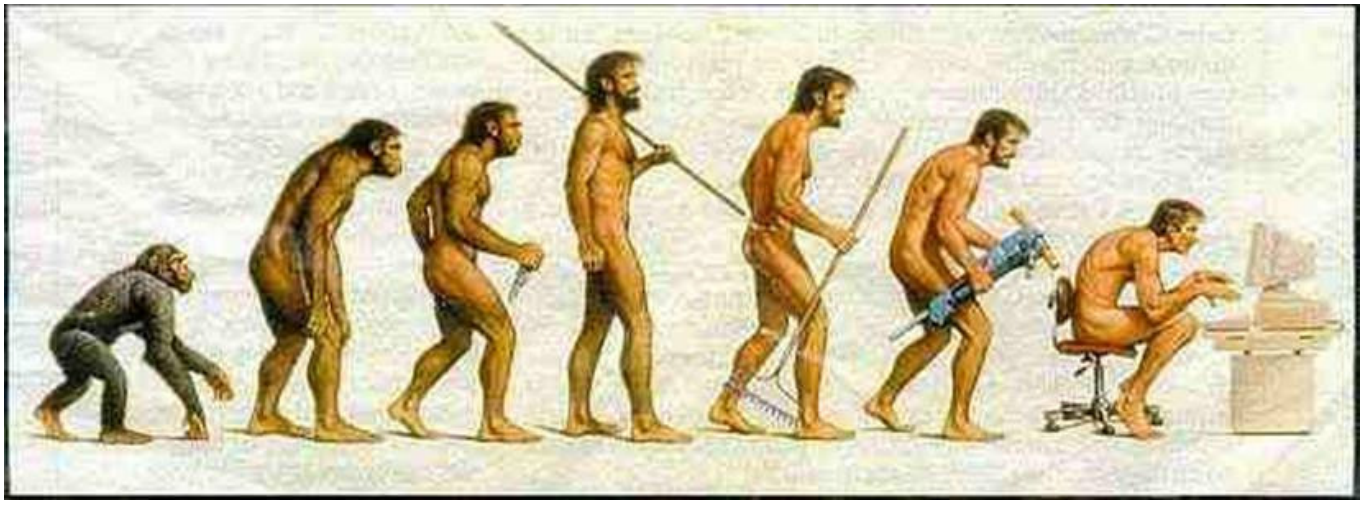


- Each brain explores a potential solution
- Instead of exchanging synaptic configurations, exchange ideas through language

(Hutchins & Hazelhurst 2005)

Memes (R. Dawkins)

Genetic Algorithms	Evolution of ideas
Population of candidate solutions	Brains
Recombination mechanism	Culture and language



R. Dawkins' *Selfish Gene*, 1982

Unsupervised Training Principles

- Maximum likelihood
- Information preservation + mutual predictibility between subsets of variables
  - Sparsity
  - Temporal constancy
- Score matching
- Matching statistics (CD-like algorithms, Max Welling's ICML 2009 paper): most flexible?

Discussion Questions

- What neuron models? Is sampling necessary?
- Why is unsupervised pre-training helping so much to train deep neural networks?
- Why is credit assignment not well carried by gradient through many layers?
- Are there general principles exploited by brains to deal with this difficult non-convex optimization?
    - Optimizing easier proxys (continuation methods)?
    - Guiding the learning of intermediate representations?

Collaborators

- Samy Bengio
- James Bergstra
- Ronan Collobert
- Aaron Courville
- Olivier Delalleau
- Dumitru Erhan
- Pascal Lamblin
- Hugo Larochelle
- Jérôme Louradour
- Pierre-Antoine Manzagol
- Pascal Vincent
- Jason Weston