

# Deep learning of representations and its application to computer vision

Ian Goodfellow

# Summary

- Deep learning background
- Four articles:
  - Spike-and-slab modeling
  - Multi-prediction deep Boltzmann machines
  - Maxout
  - Street number transcription

# Machine learning

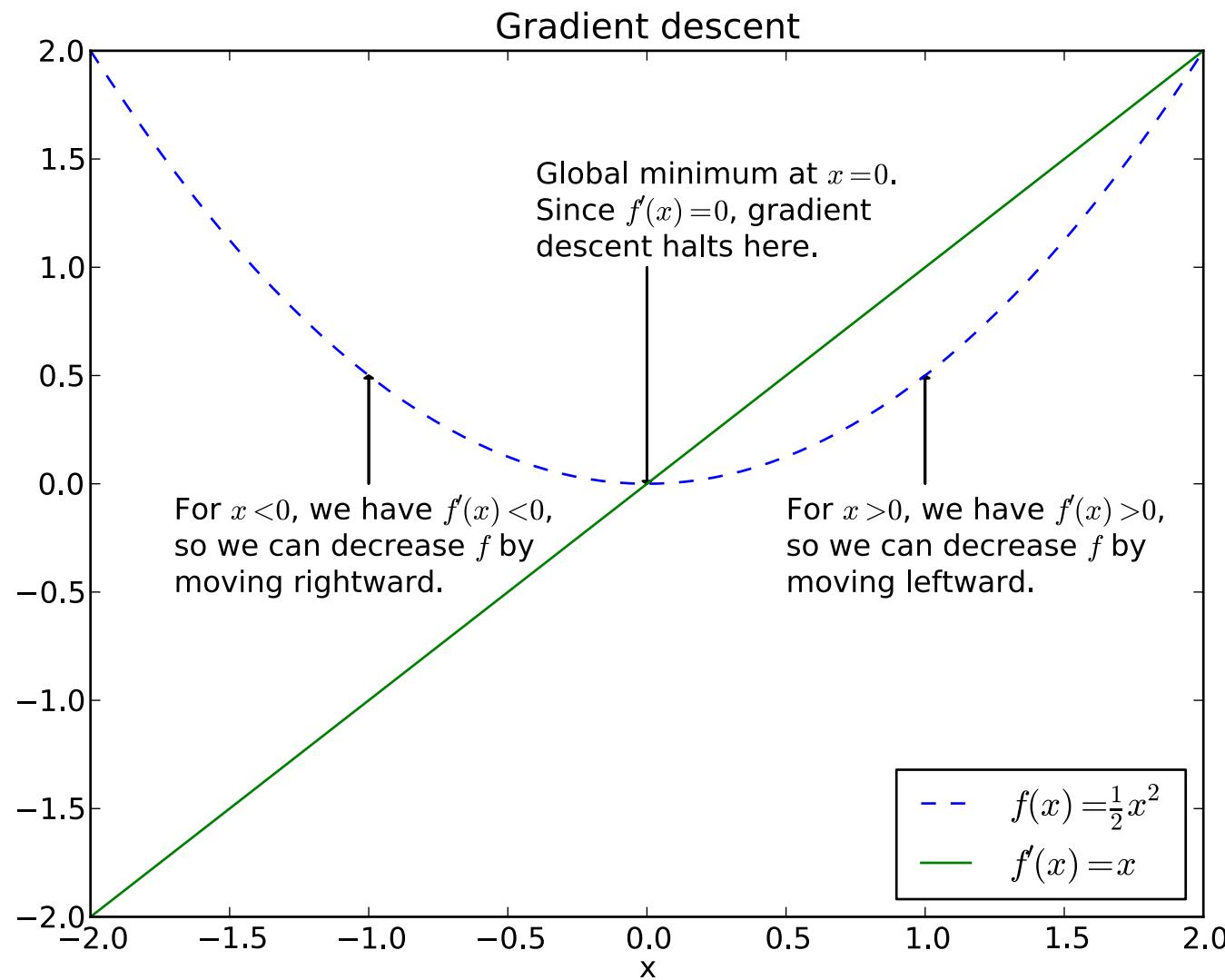
“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

-Tom Mitchell

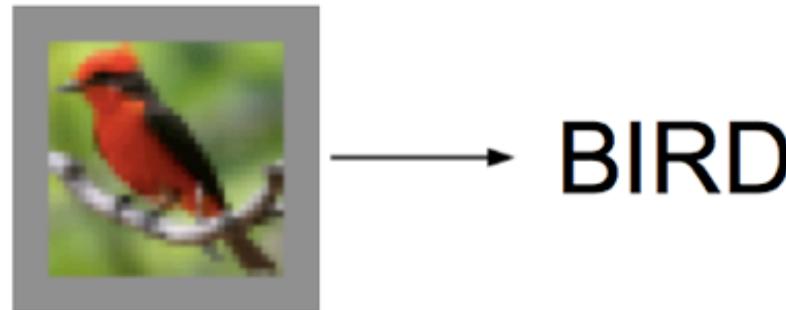
# Maximum likelihood estimation

- Pick parameters that maximize model's probability of generating the observed data
- Given enough data, recovers the true model

# Gradient descent

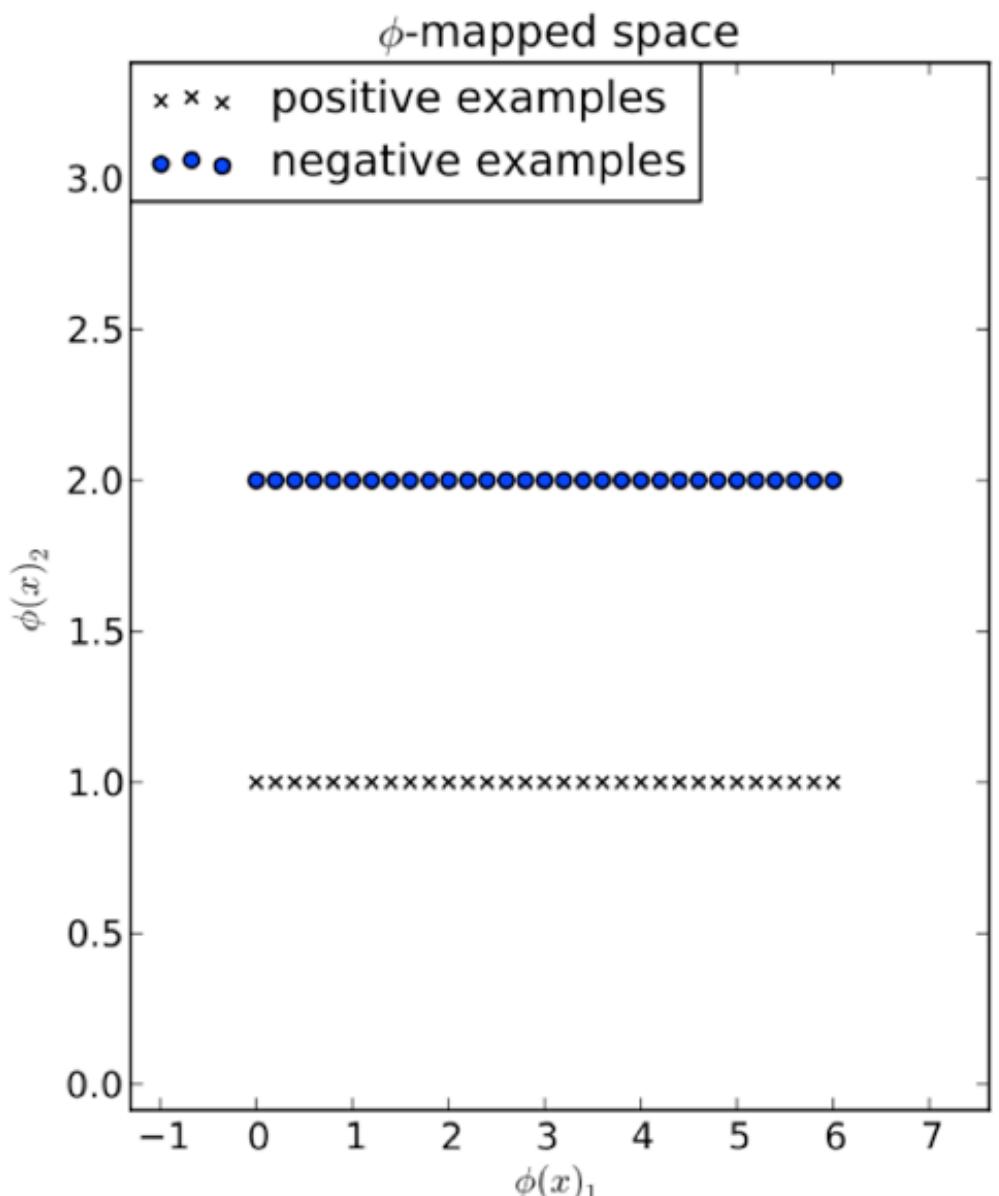
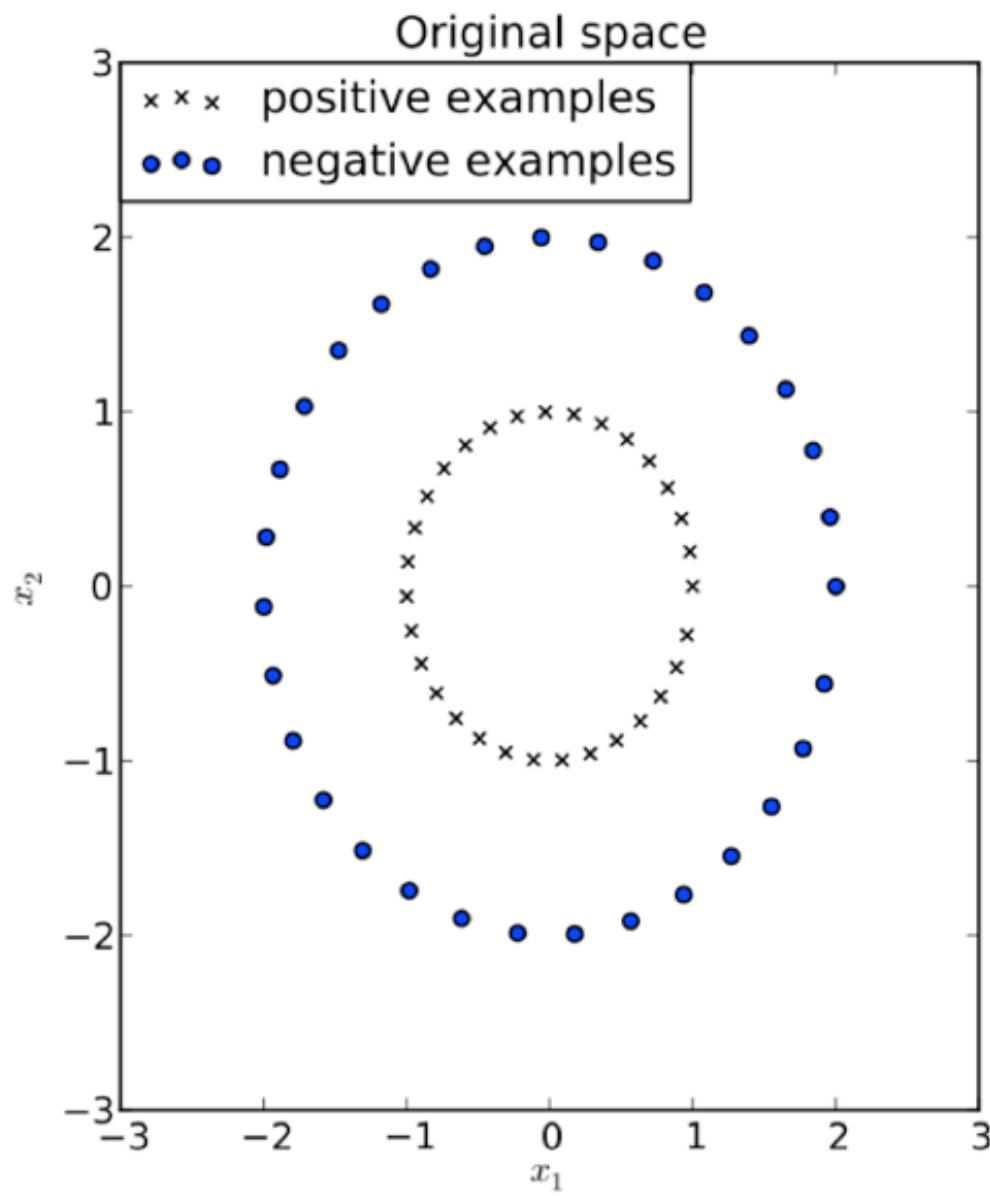


# Supervised Learning

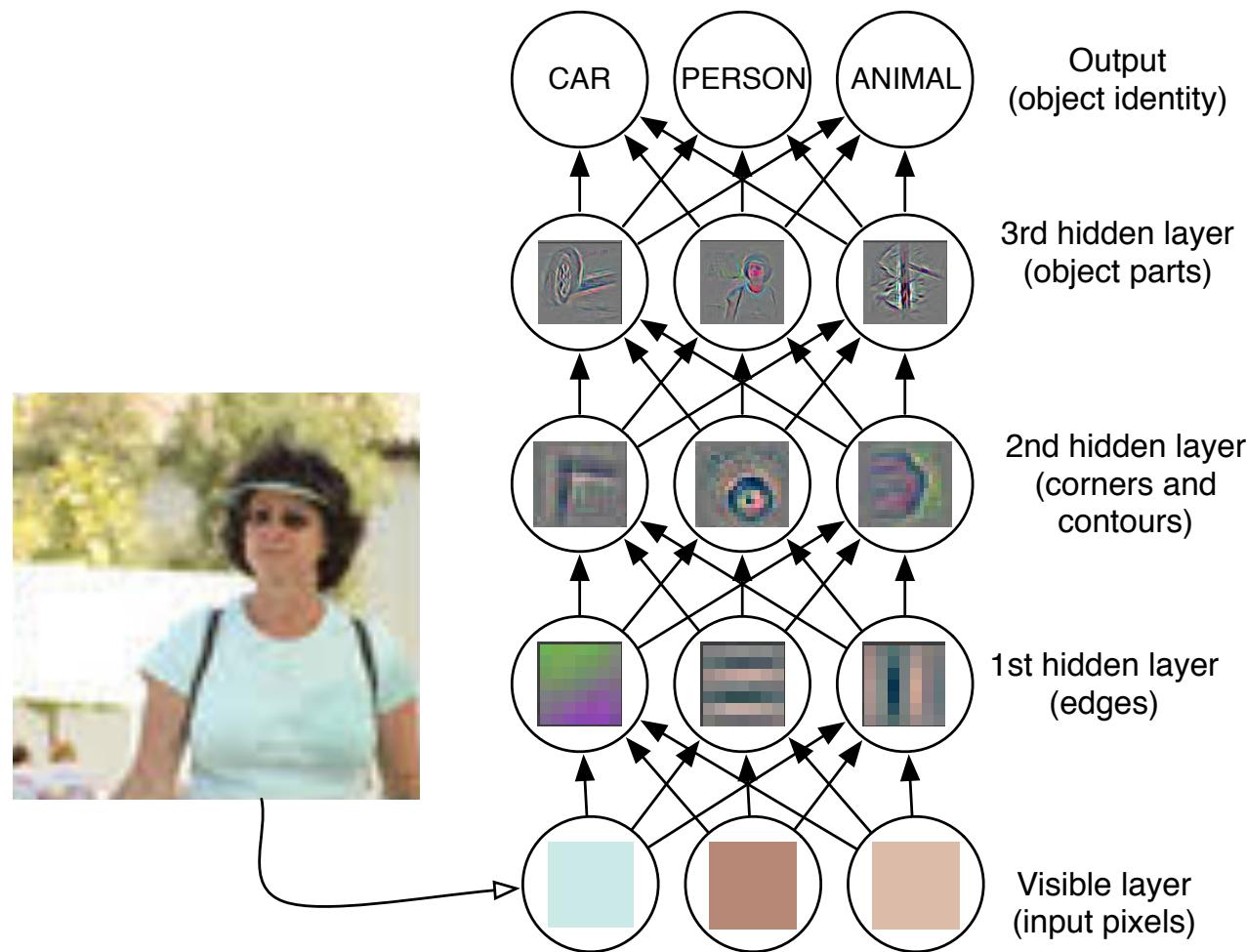


- Data is features  $X$  and targets  $y$
- Goal: learn to map  $x$  to  $y$
- Classification: discrete  $y$
- Regression: continuous  $y$

# Unsupervised learning for feature learning



# Deep learning



(Images by Matt Zeiler)

# Spike-and-Slab Sparse Coding

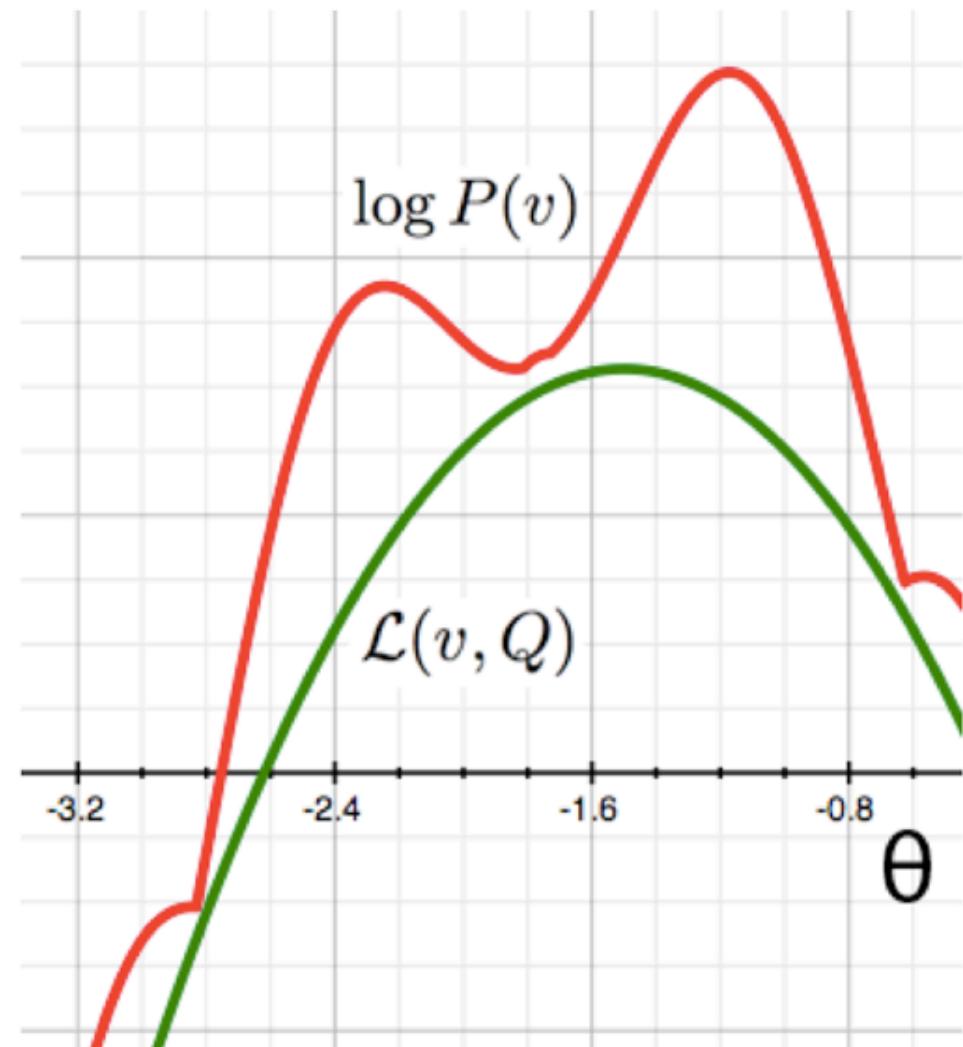
- Co-authors: Aaron Courville and Yoshua Bengio
- Motivated by Adam Coates' work on feature learning and feature extraction
- Faster form of variational inference
- Component of a deep model

# Motivating CIFAR-10 results

- Validation set accuracy (Coates and Ng 2011):
  - RBM features encoded with RBM: 74.1%
  - RBM features encoded with sparse coding: 76.7%
- Test set accuracy (Courville et al 2011):
  - ssRBM: 76.7%

# Variational learning

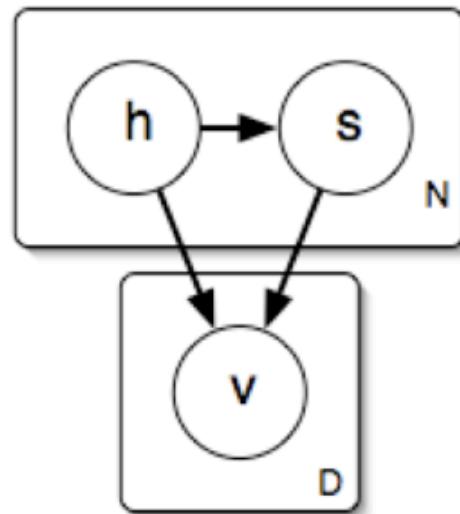
- Approximate intractable  $P(h|v)$  with tractable  $Q(h)$
- Use  $Q$  to construct a lower bound on the log likelihood



# Variational inference

- $\mathcal{L}(v, Q) = \mathbb{E}_{h \sim Q}[\log P(v, h)] + H_Q(h)$
- Maximizing this corresponds to minimizing  
$$D_{KL}(Q(h) \| P(h | v))$$
- Often requires both analytical and iterative optimization

# The Spike-and-Slab Sparse Coding (S3C) Generative Model

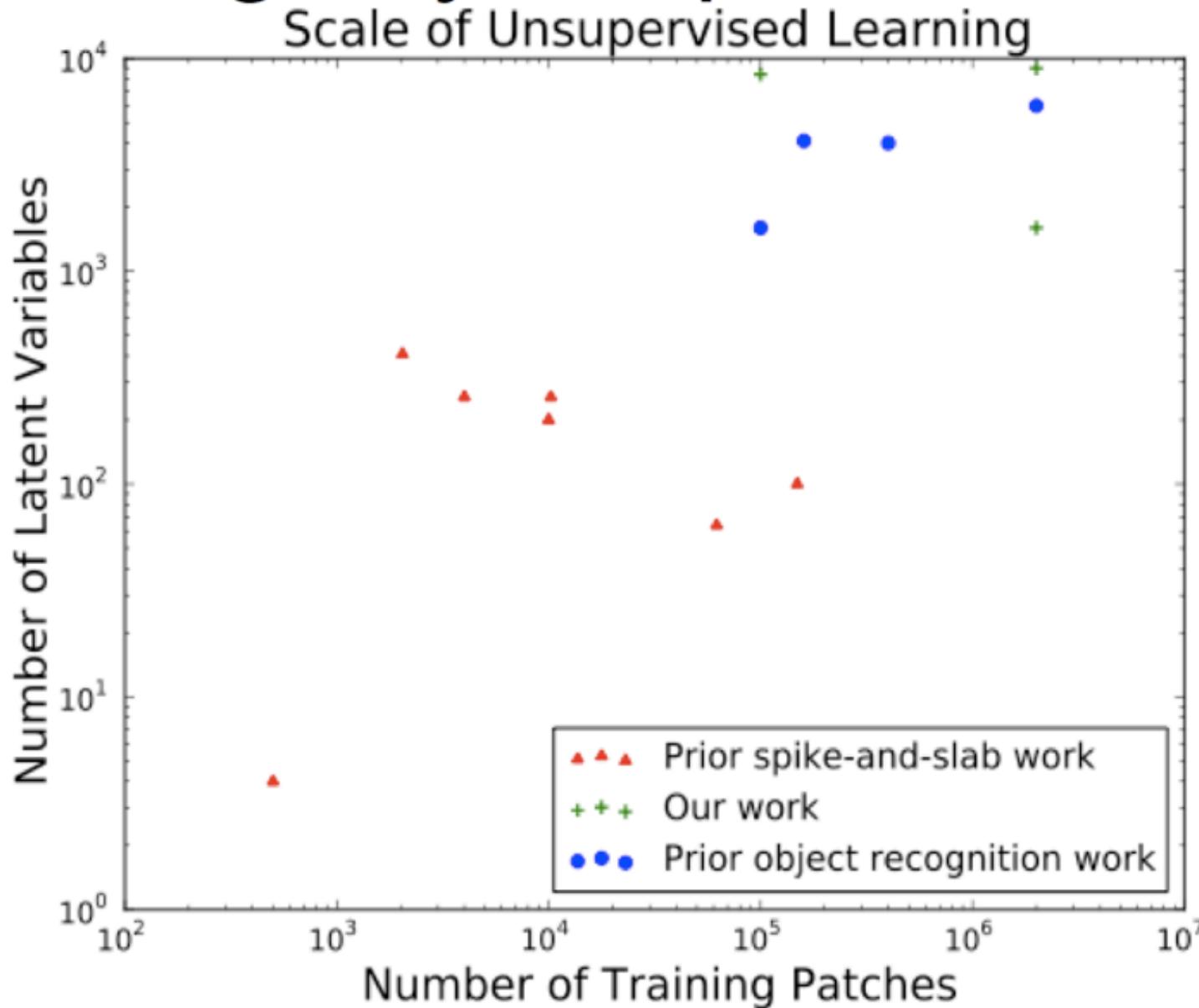


$$p(h_i = 1) = \sigma(b_i)$$

$$p(s_i | h_i) = \mathcal{N}(s_i | h_i \mu_i, \alpha_{ii}^{-1})$$

$$p(v_d | s, h) = \mathcal{N}(v_d | W_{d:}(h \circ s), \beta_{dd}^{-1})$$

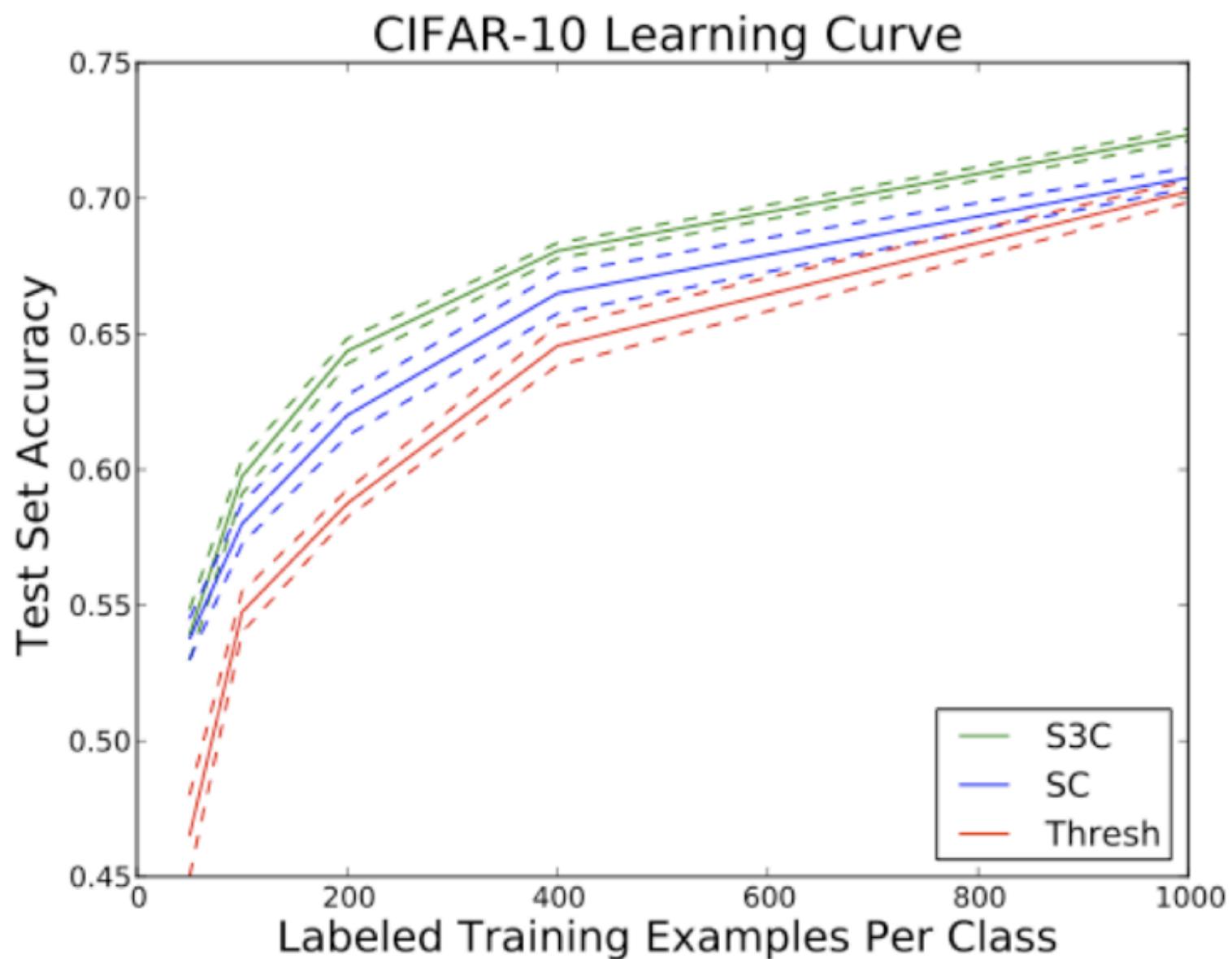
# Scaling beyond previous work



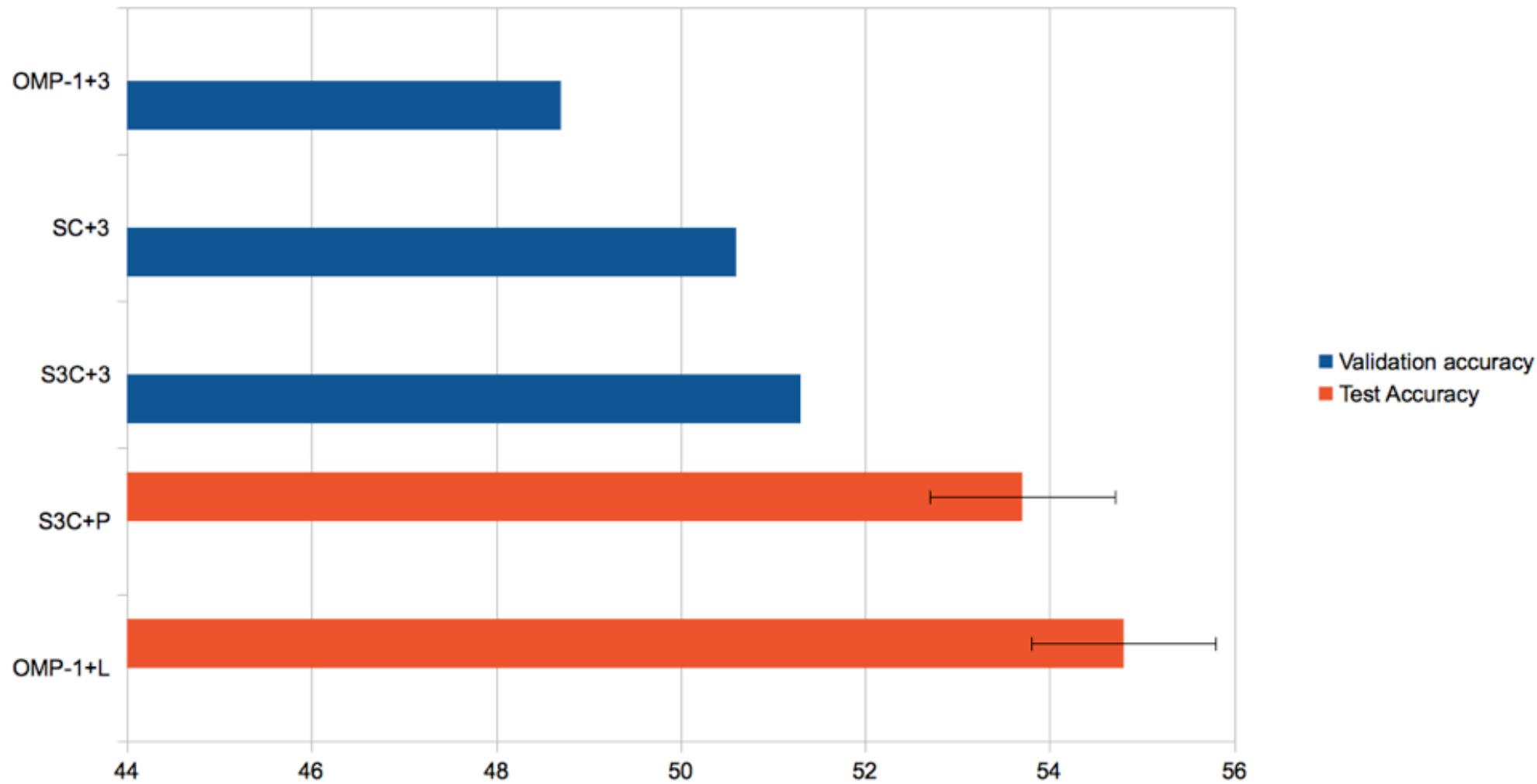
Spike-and-Slab work: Mohammed et al, 2011; Zhou et al., 2009; Garrigues and Olsahusen, 2008; Lücke and Sheik, 2011; Titsias and Lázaro-Gredilla, 2011

# Scaling to more classes and fewer labeled examples per class

- Clic



# CIFAR-100 Results



# Transfer Learning Challenge

Labeled training set:



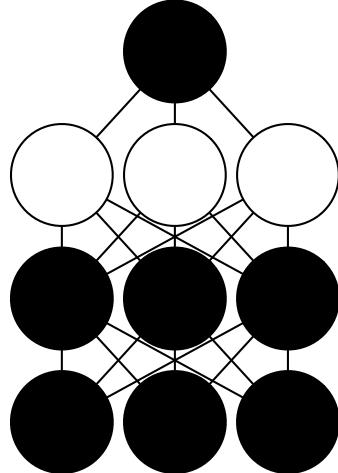
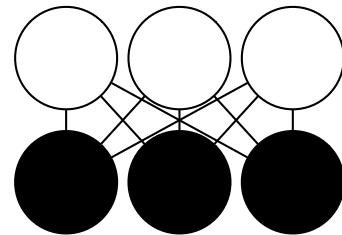
Self-taught learning with S3C won  
the challenge with a test set  
accuracy of 48.26%

# Multi-Prediction deep Boltzmann machines

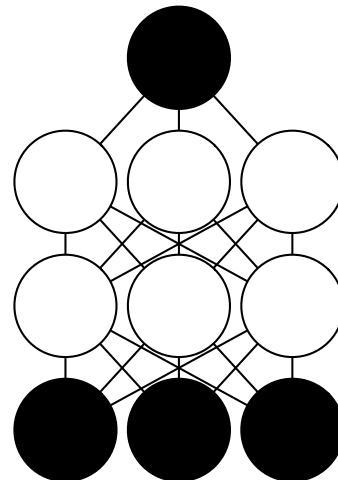
- Co-authors: Mehdi Mirza, Aaron Courville, Yoshua Bengio
- Simplified training procedure for deep Boltzmann machines
- Improved accuracy of approximate inference

# Typical DBM Training

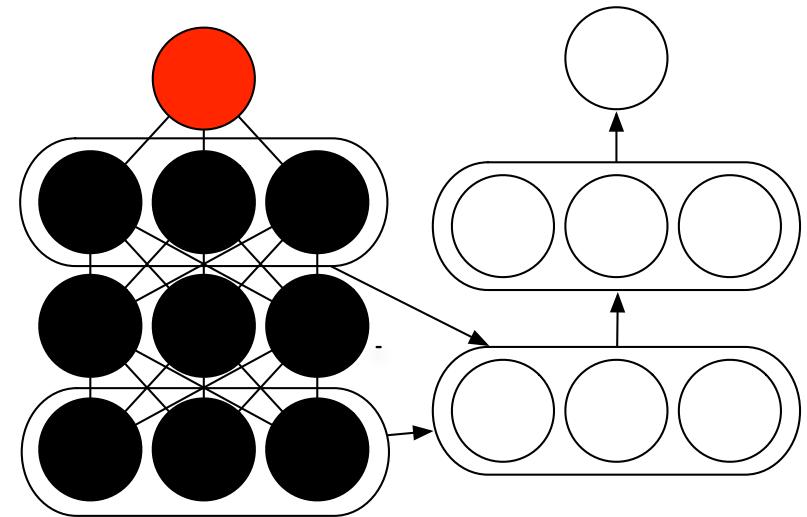
1. Greedy  
layerwise  
pretraining



2. Joint  
generative  
training



3. Discriminative  
fine-tuning



# Sampling-based approximations

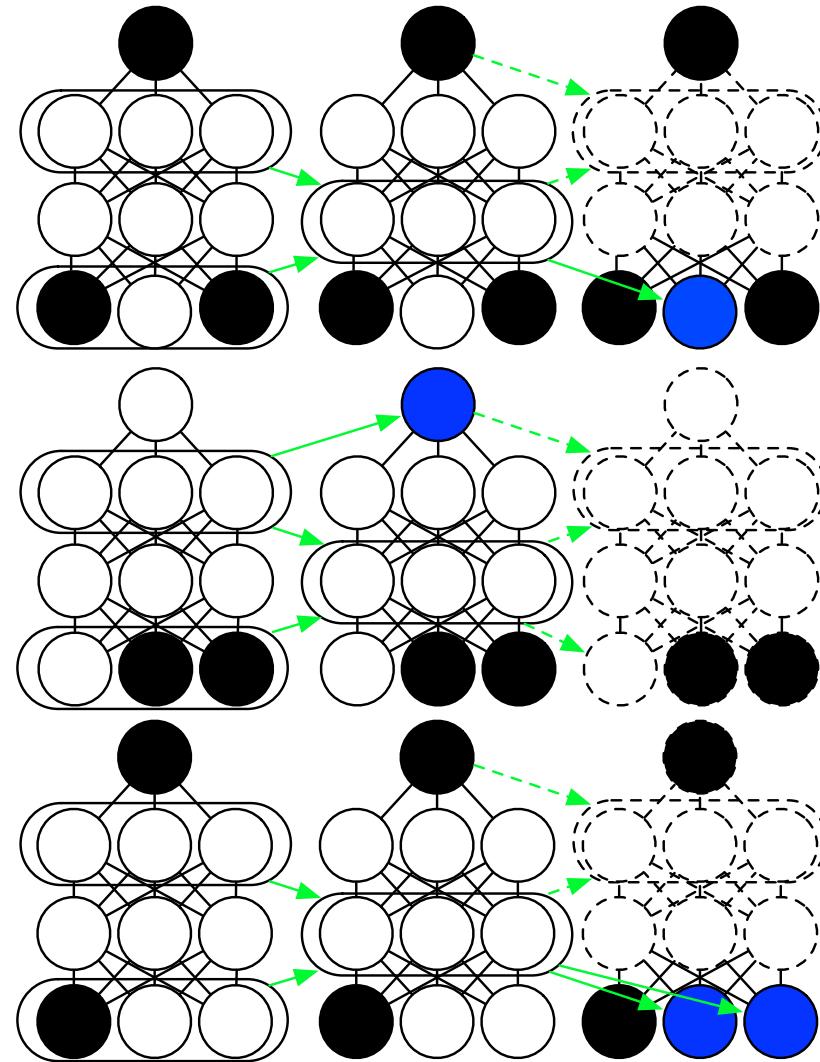
- $p(x; \theta) = \exp(-E(x; \theta)) / Z(\theta)$
- What if  $Z(\theta)$  is intractable?
- $\frac{\partial}{\partial \theta_i} \log Z = -\mathbb{E}_x \left[ \frac{\partial}{\partial \theta_i} E(x; \theta) \right]$
- Approximate expectations via sampling
- CD-k: sample k steps from data points
- SML/PCD: sample continuously, use low learning rate

# Simplify, simplify, simplify

	Classic approach	Goal
# models	$\# \text{layers} + 2$	1
# criteria	$\# \text{layers} + 2$	1
Classifier	Extra classifier model	Same unified probabilistic model

# Multi-Prediction Training

Randomly  
sample  
different  
inference  
problems



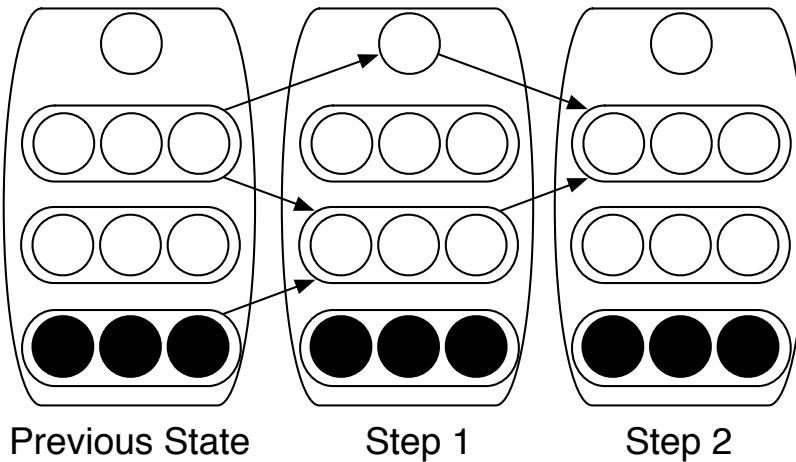
Backprop  
through the  
mean field  
inference  
graph

# Benefits of Multi-Prediction Training

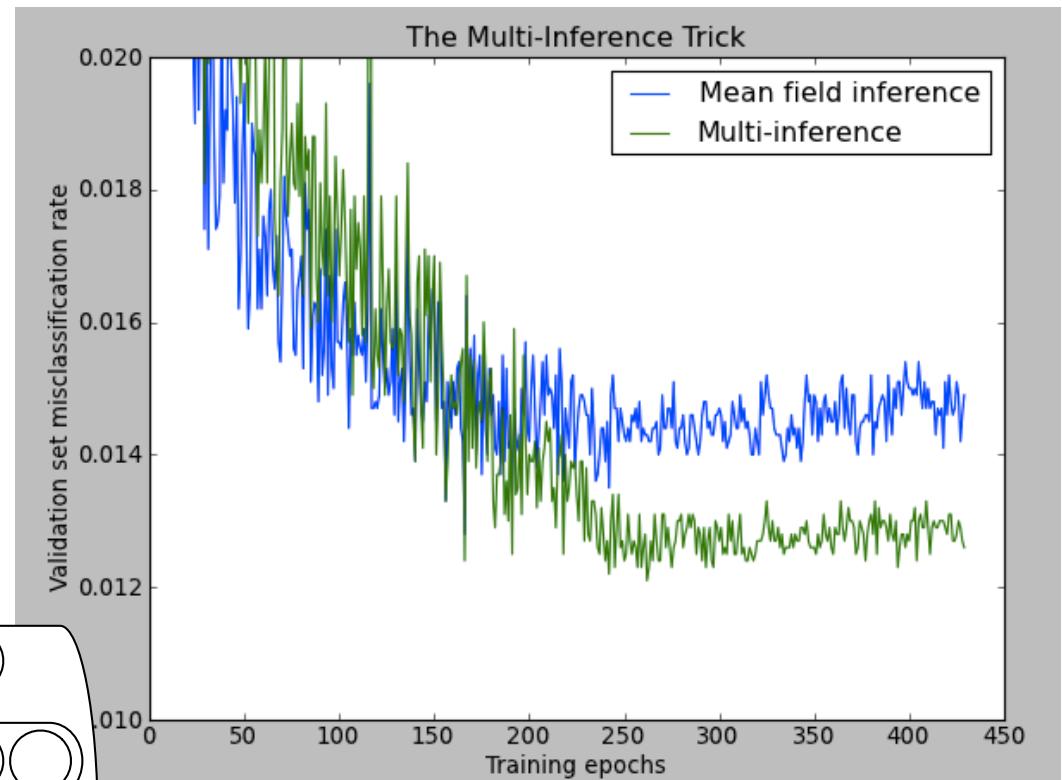
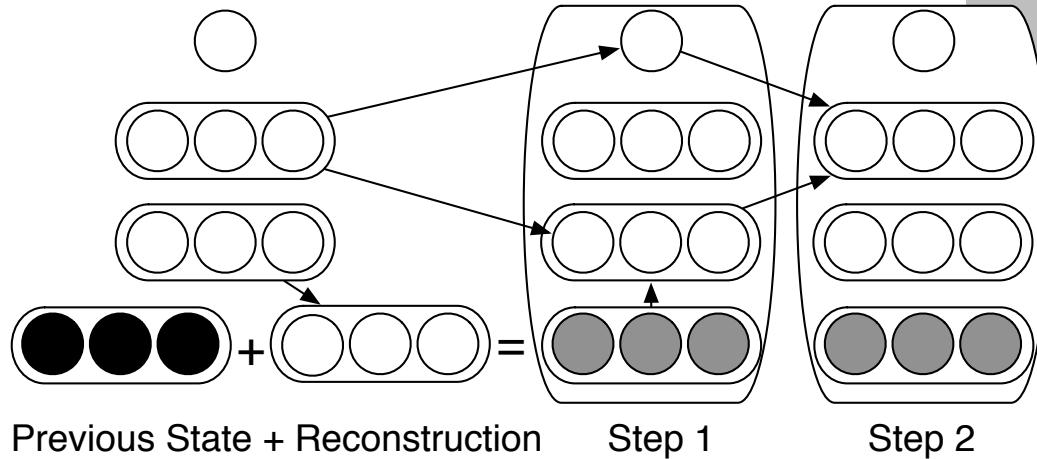
- Learning rate doesn't affect approximation accuracy
- Training compensates for approximate inference
- Similar to Stoyanov et al 2011

# Multi-Inference Trick

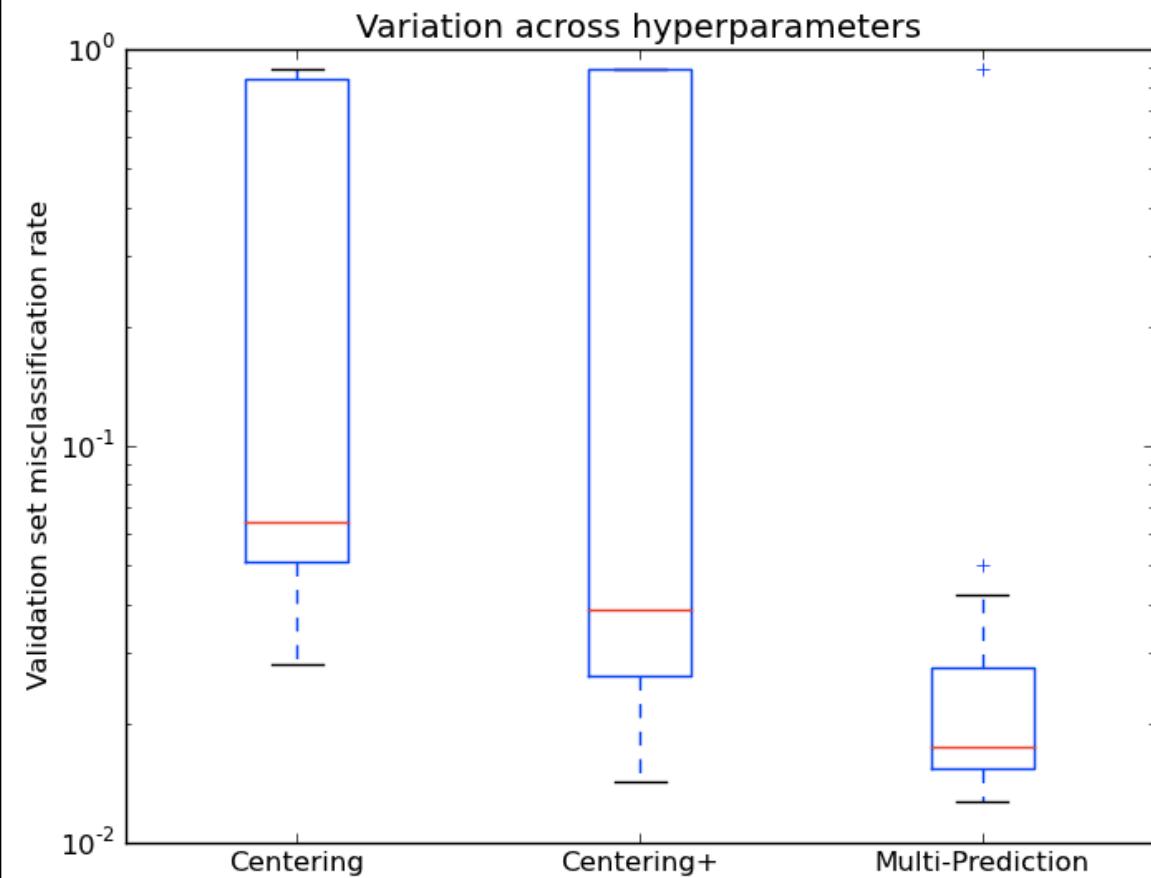
## Mean Field Iteration



## Multi-Inference Iteration



# Results



Model	Test error with fine-tuning
S&H 2009*	0.95
Centered DBM	1.22
MP-DBM	0.99

Centering: Montavon and Müller, 2012

Multi-prediction, 2X hidden units, no fine-tuning\*: 0.91

\*Retrained using validation set.

# Mission Accomplished

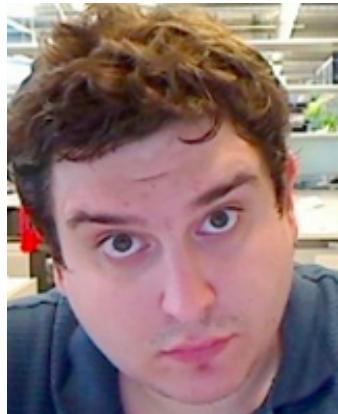
	Classic approach	Goal
# models	$\# \text{layers} + 2$	1 ✓
# criteria	$\# \text{layers} + 2$	1 ✓
Classifier	Extra classifier model	Same unified probabilistic ✓ model

# Maxout Networks

by Ian Goodfellow

Joint work with

David Warde-Farley



Mehdi Mirza



Aaron Courville



Yoshua Bengio



with acknowledgments to

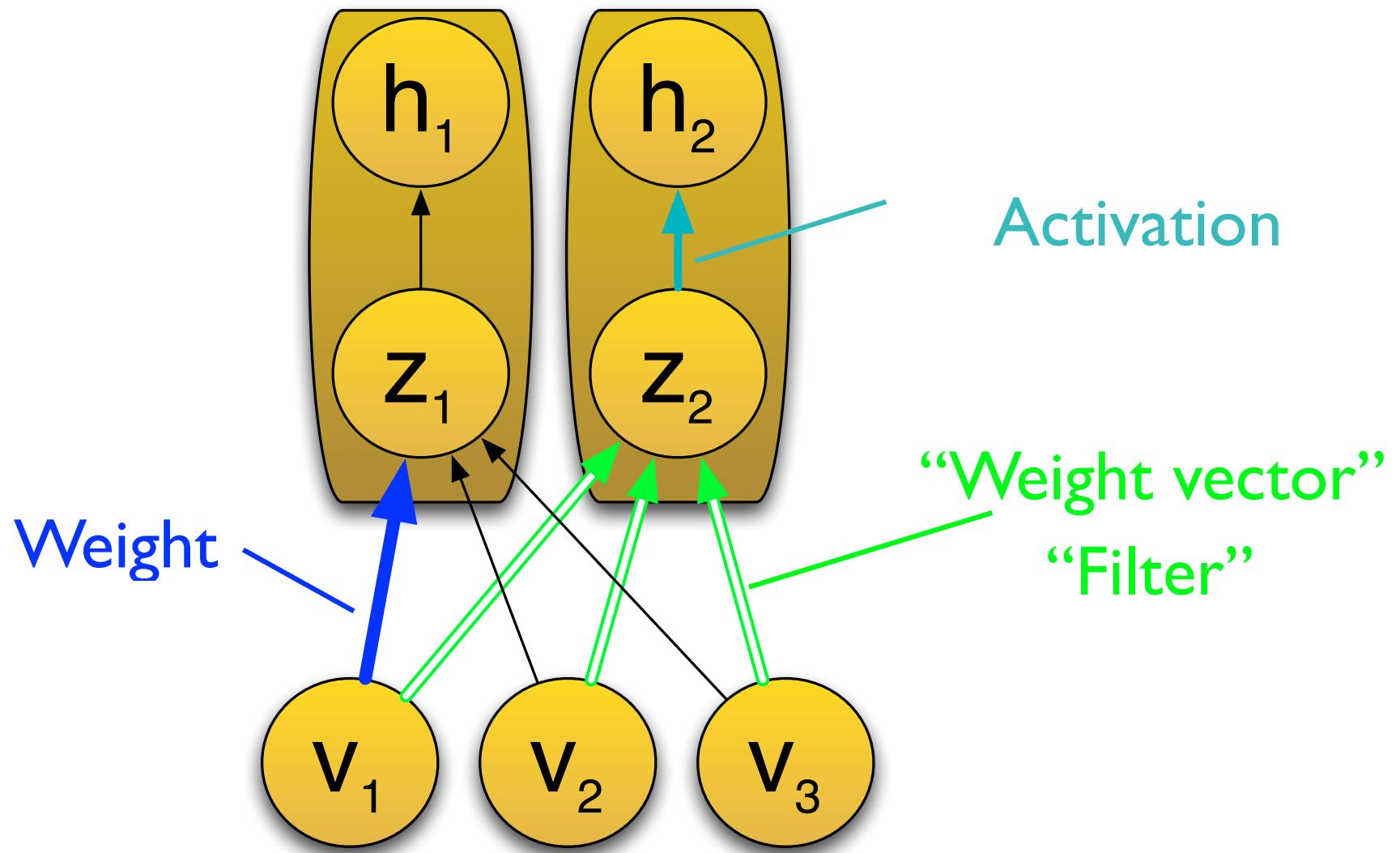


Frédéric Bastien

Yann Dauphin

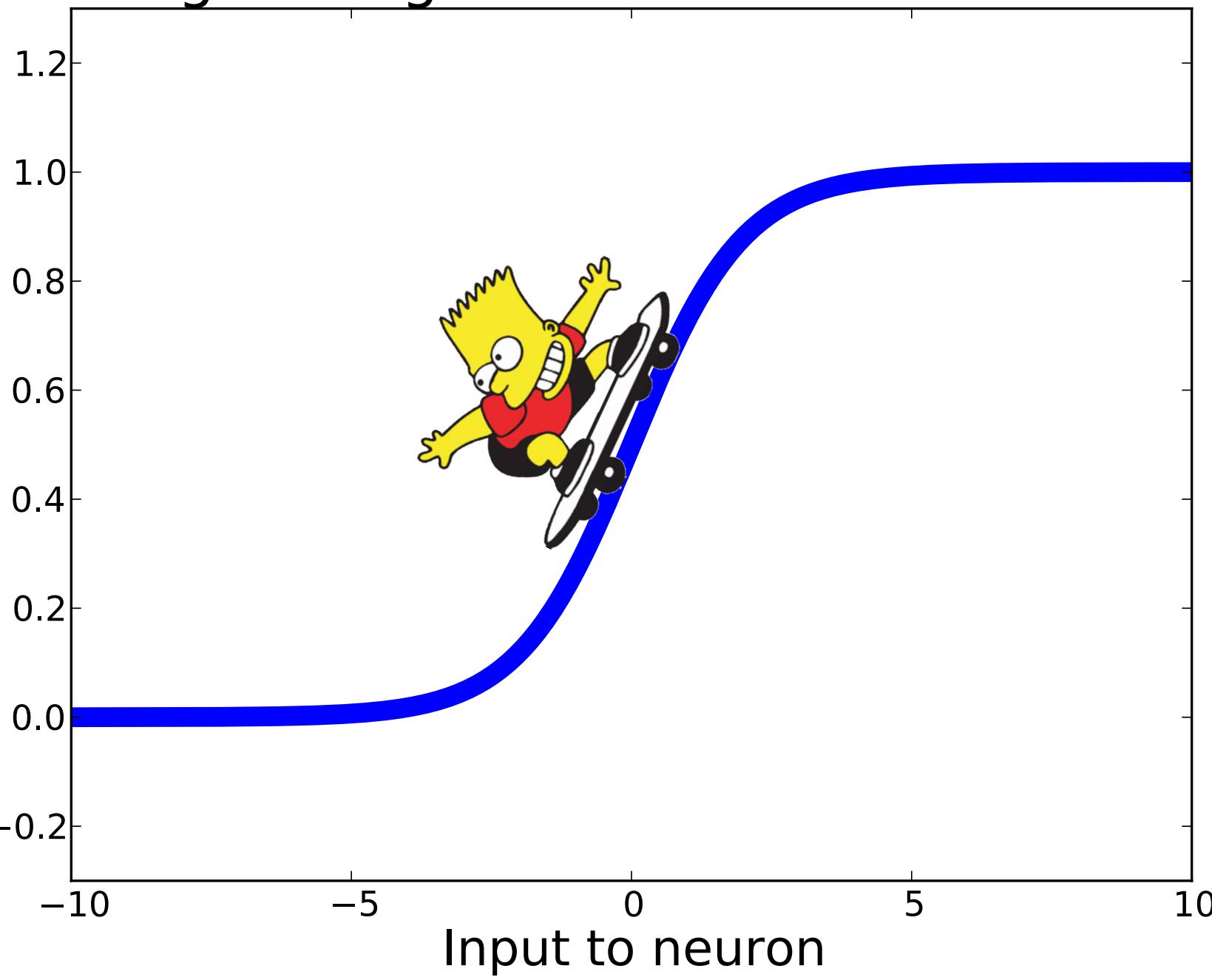
Pascal Lamblin

# Traditional activation functions

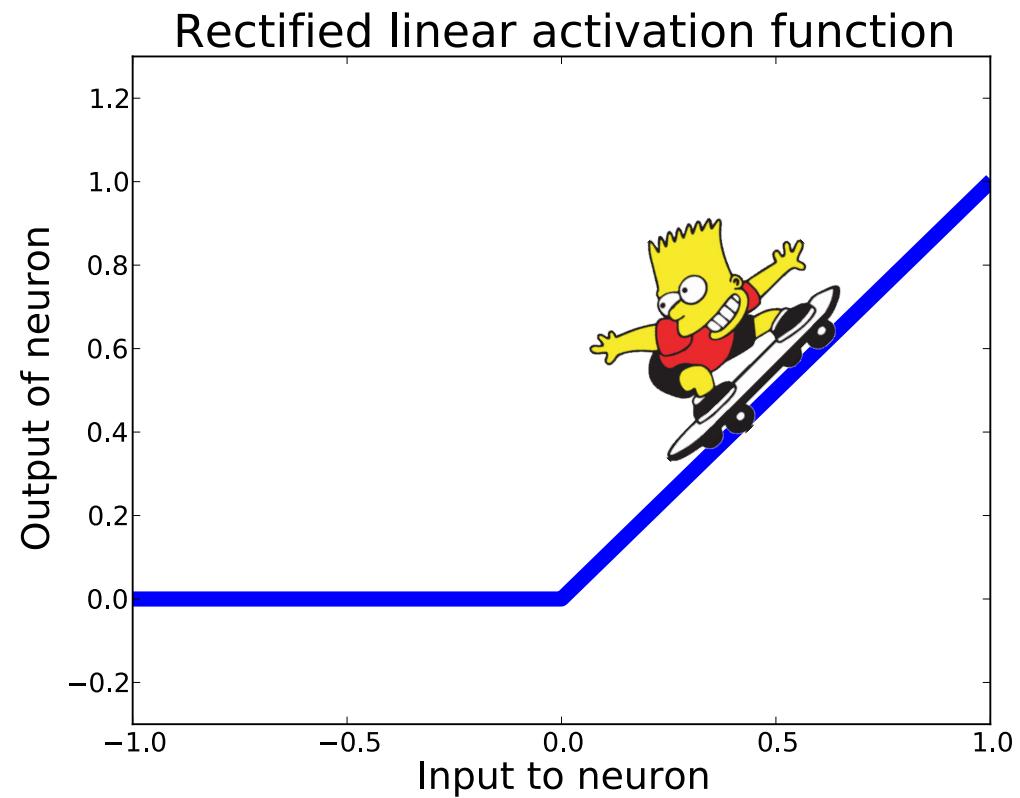
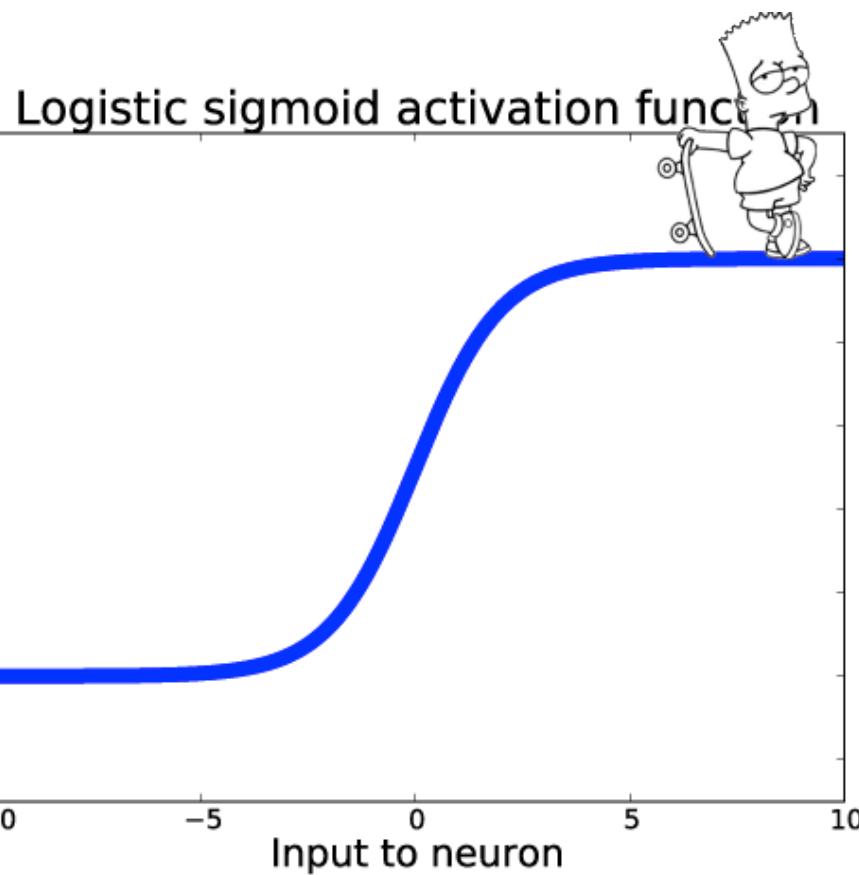


# Logistic sigmoid activation function

Output of neuron

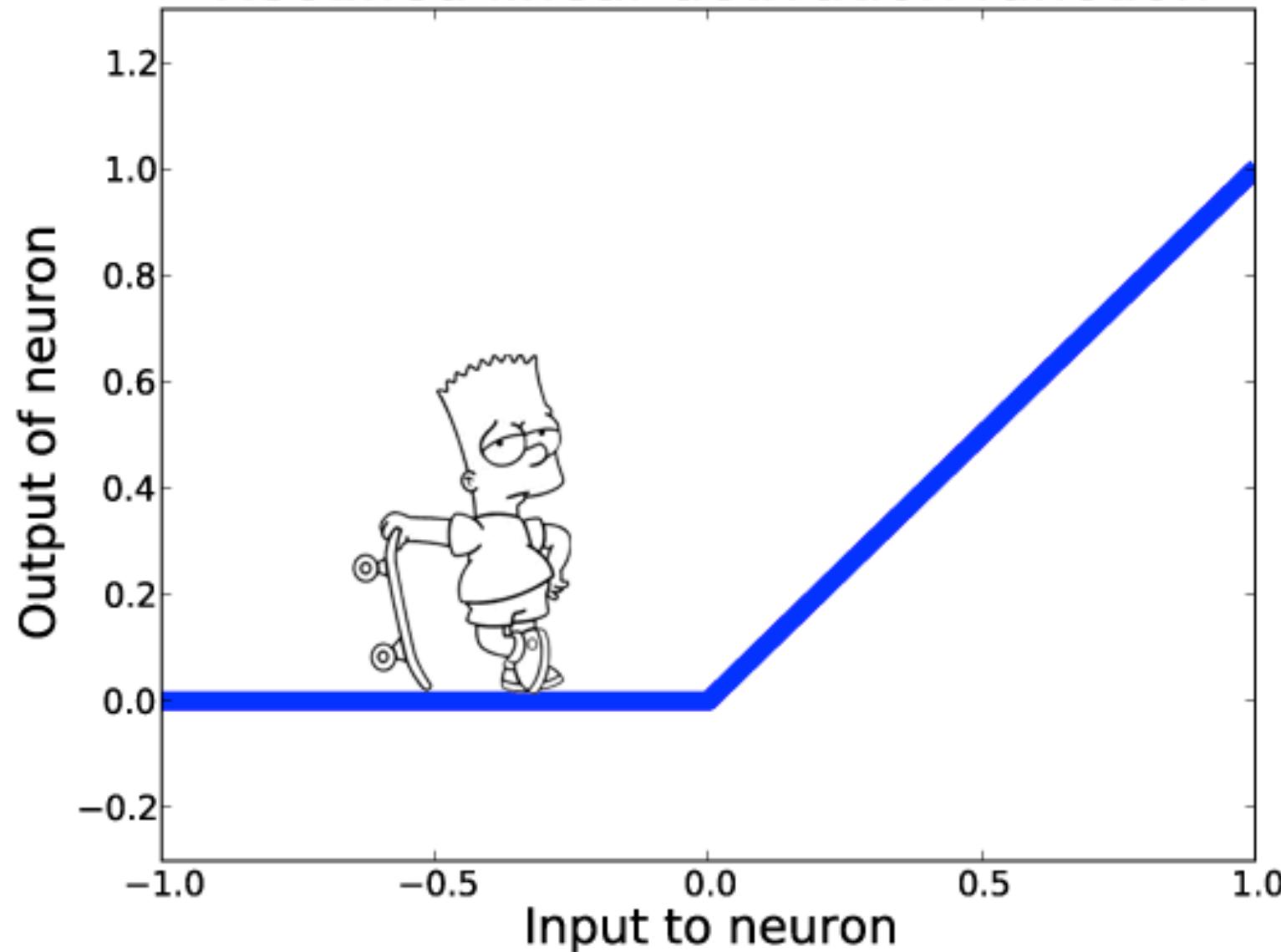


# The vanishing gradient problem

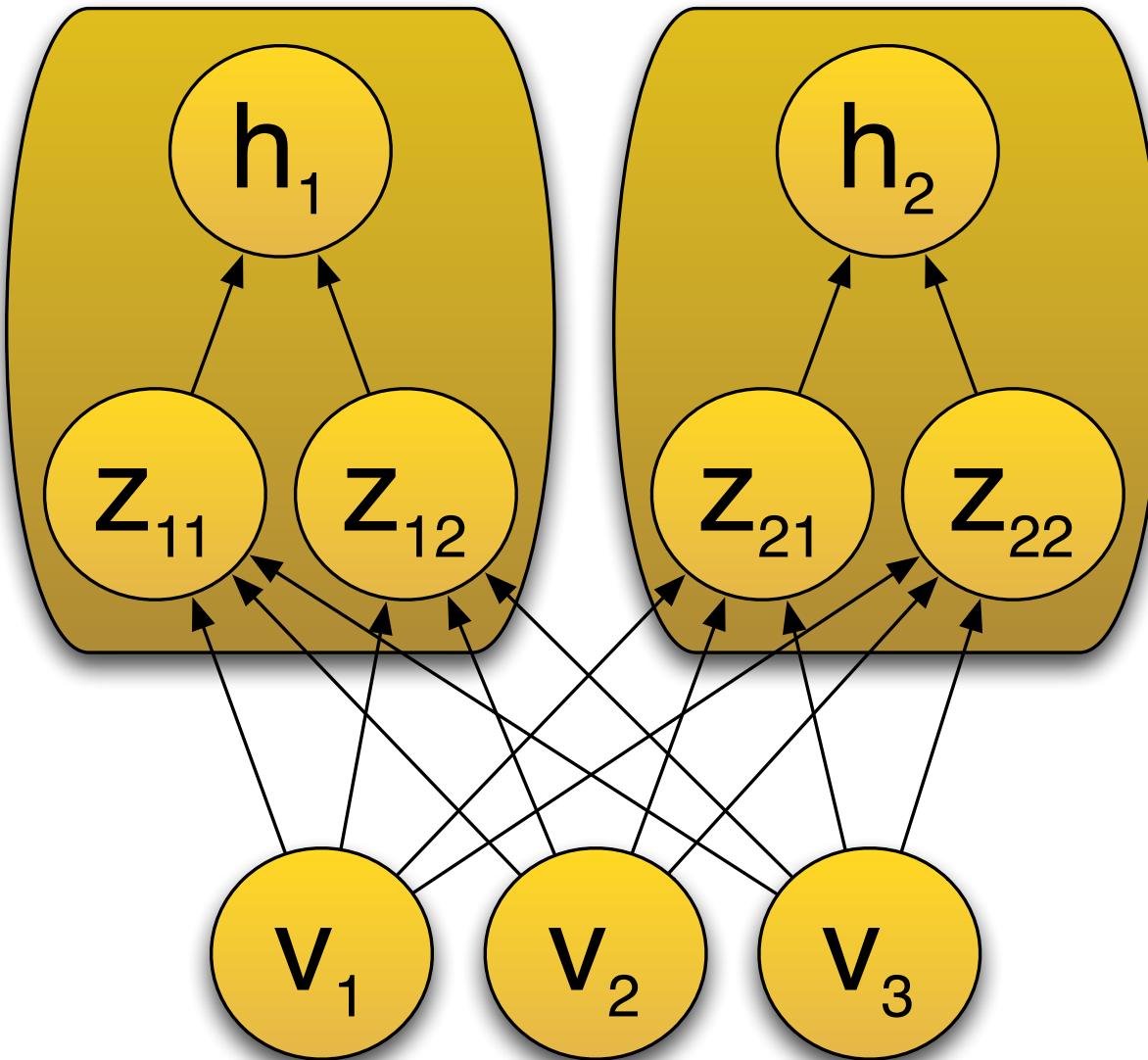


# Uh-oh

Rectified linear activation function

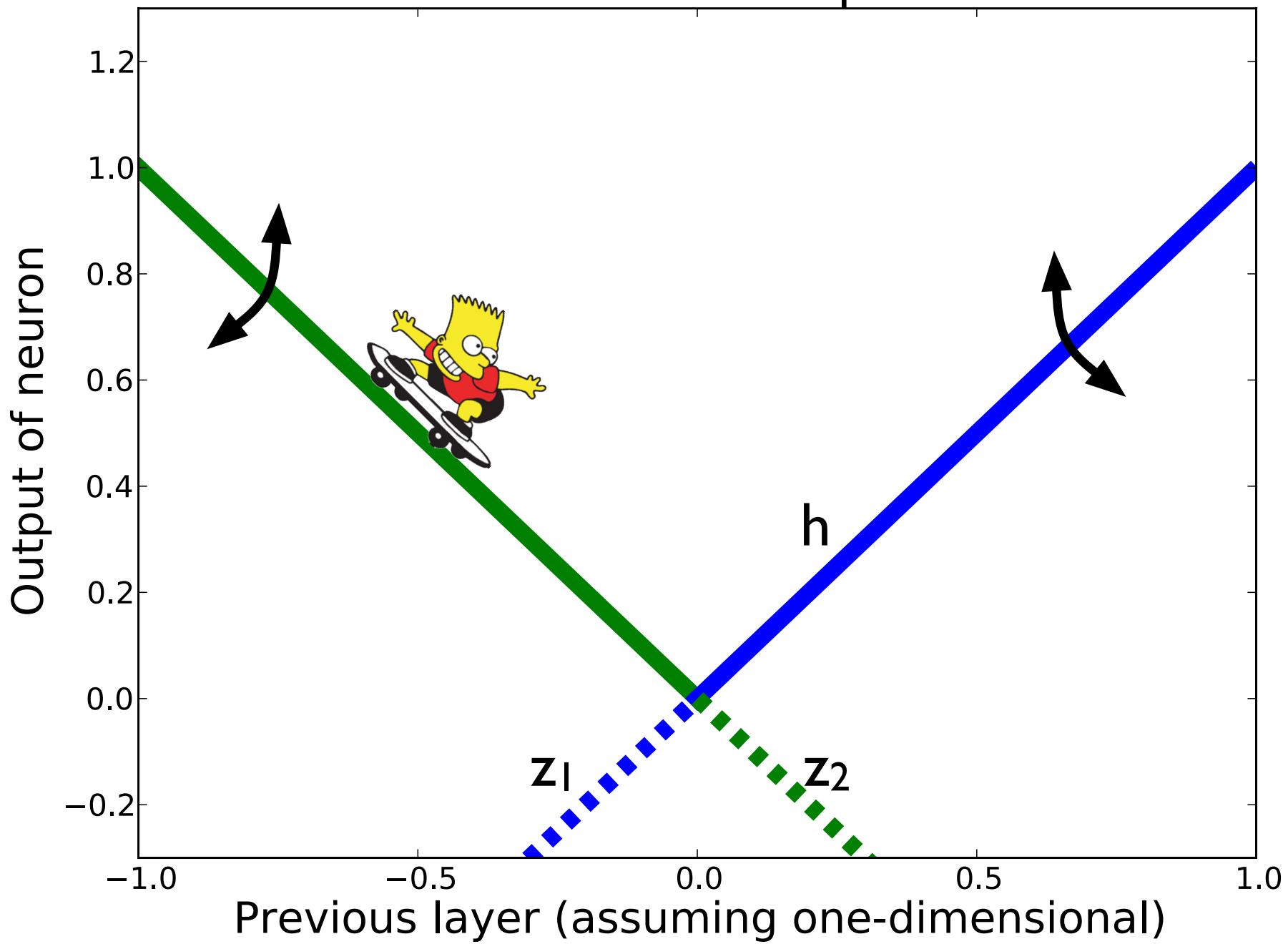


# Maxout

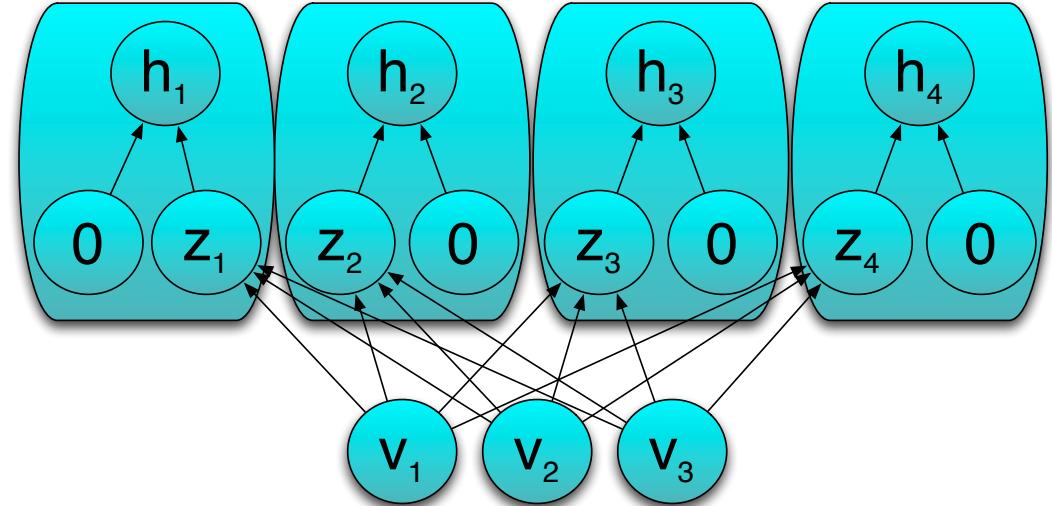
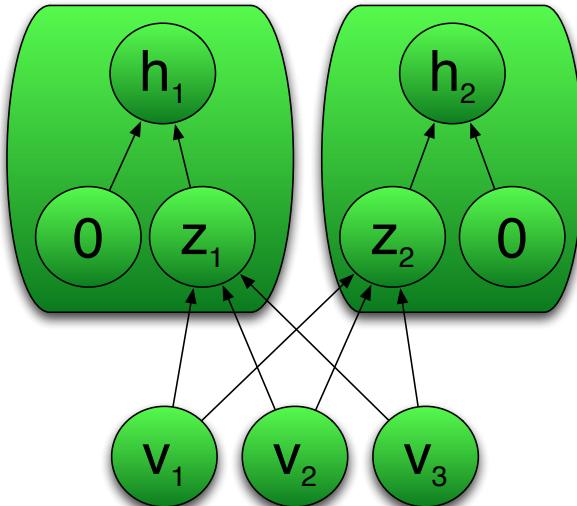
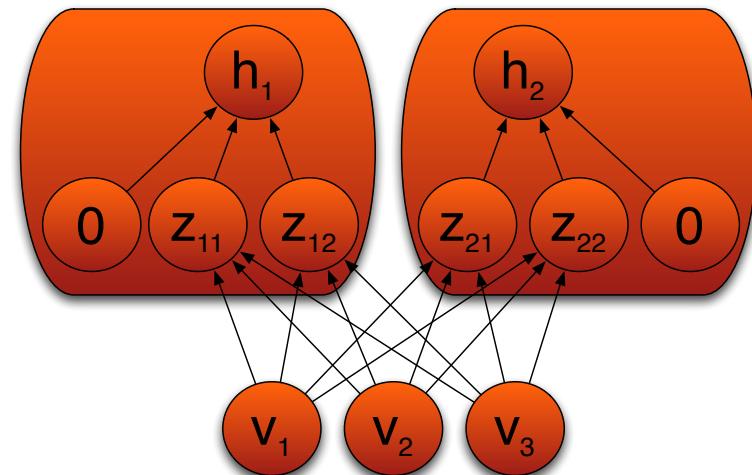
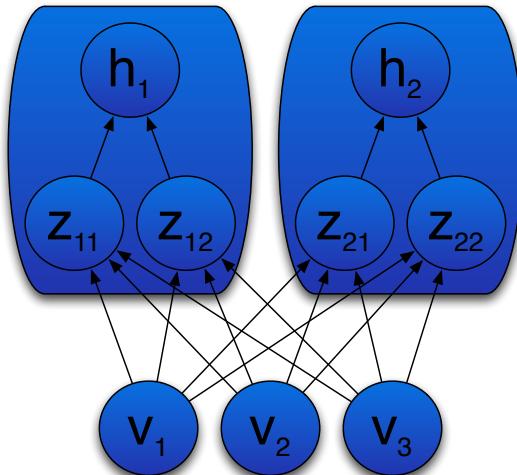


$$h_i = \max_j z_{ij}$$

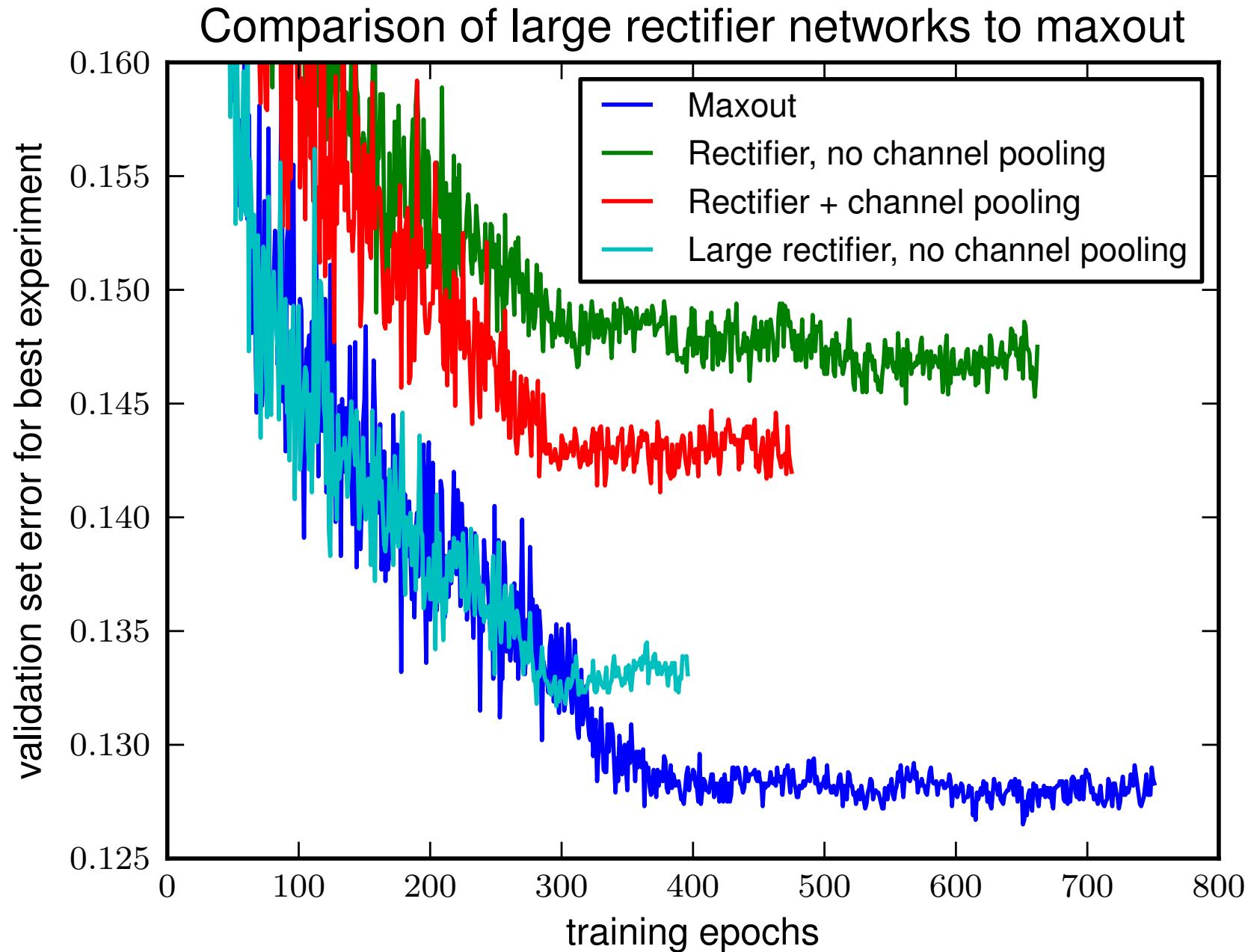
# Maxout example



# Comparing maxout to rectifiers



# Effectiveness of pooling



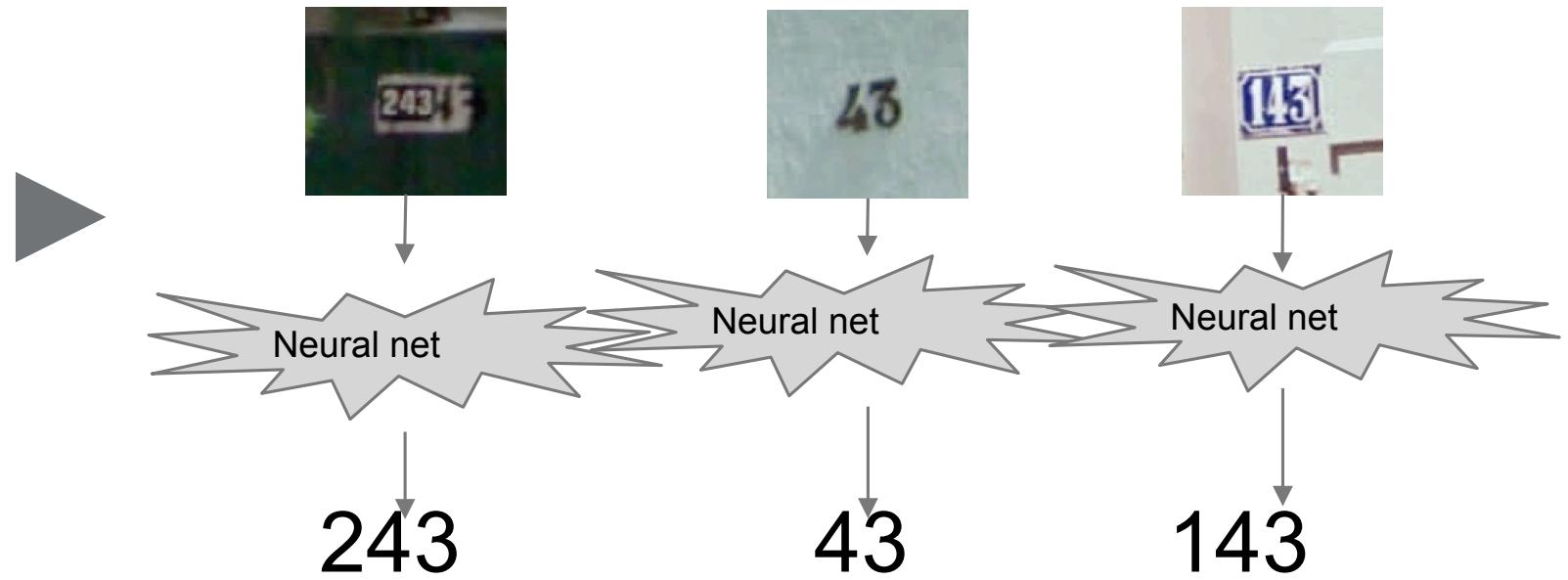
# Applications of maxout

- Speech: Miao et al 2013, Cai et al 2013, Zhang et al 2014, Swietojanski et al 2014
- Multiplayer game matchmaking: Laufer et al, 2013
- Text detection: Jaderberg et al 2014
- Text transcription: Alsharif and Pineau, 2013
- Simplifying optimization: Gulcehre 2013
- Recurrent networks: Pascanu 2014
- Whale call detection: Smirnov 2013
- Black-box classification: Xie et al, 2013

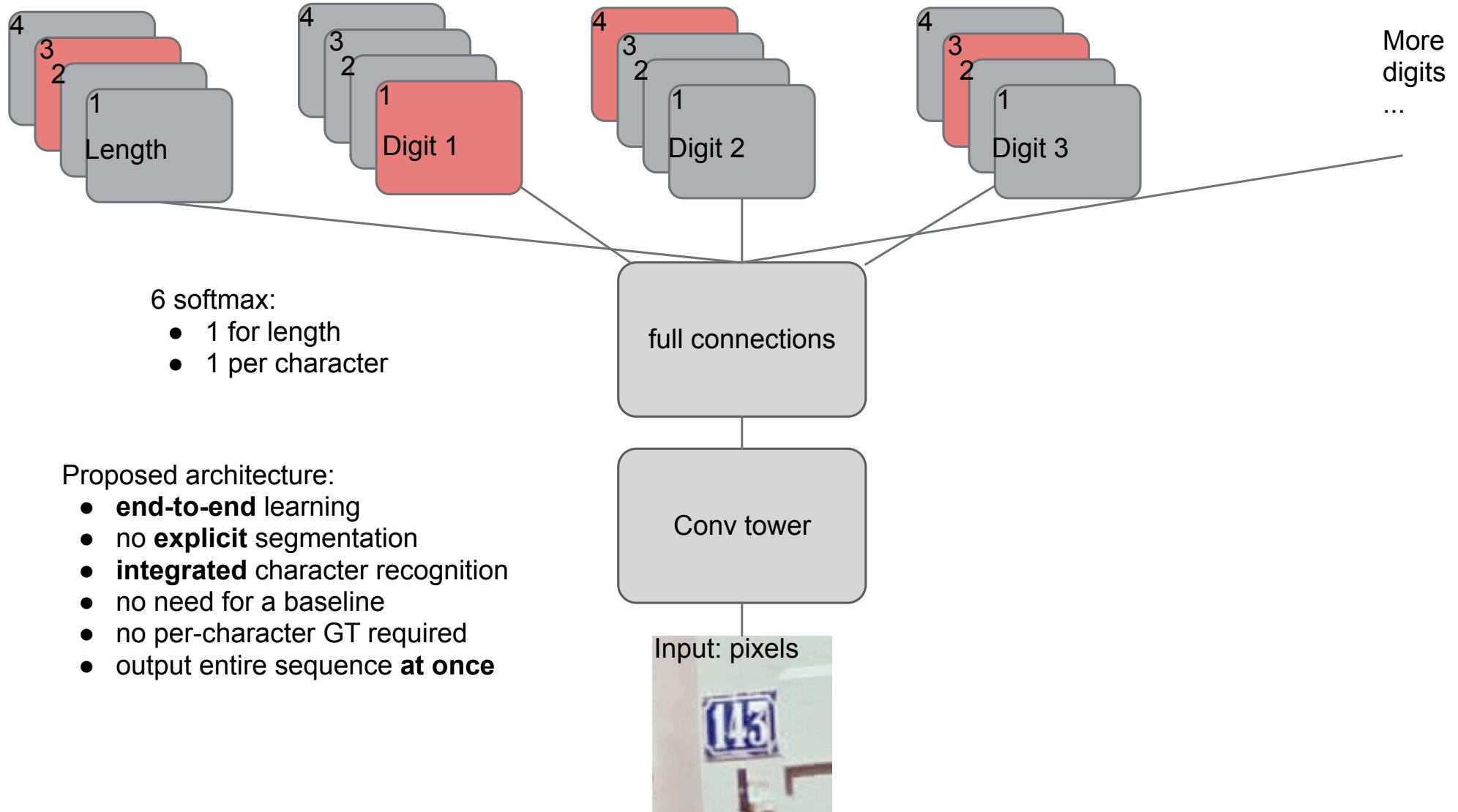
# Street number transcription

- Co-authors: Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet
- Use convolutional networks to read address numbers from Street View Images
- Automated transcription of over 100 million real address numbers

Want



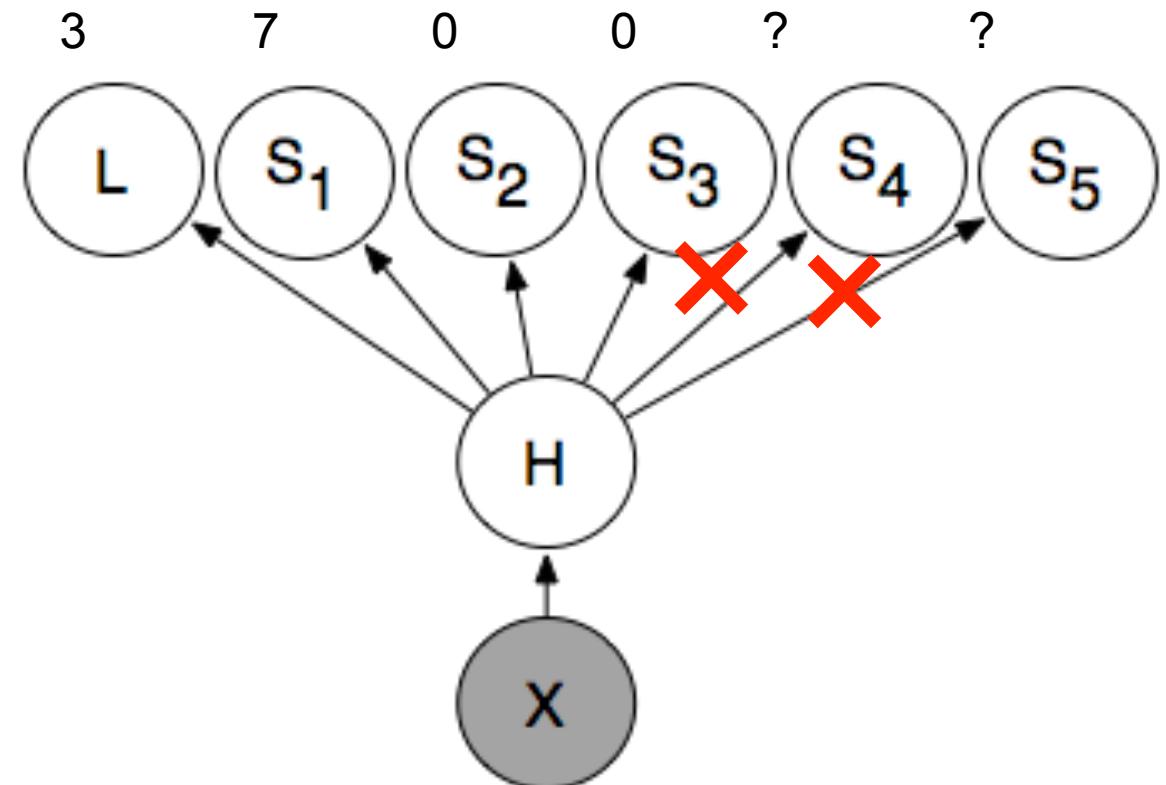
# Architecture



## Training

Log likelihood:

$$\log P(L = l \mid X) + \sum_{i=1}^L \log P(S_i = s_i \mid X)$$



## MAP sequence inference

$$\operatorname{argmax}_{L, S_1, \dots, S_L} \log P(S \mid X)$$



$$\log P(L = 1) + \log P(S_1 = "1")$$



$$\log P(L = 2) + \log P(S_{1:2} = "17")$$



$$\log P(L = 3) + \log P(S_{1:3} = "175")$$



$$\log P(L = 4) + \log P(S_{1:4} = "1751")$$



## Accuracy

	Coverage@ human accuracy (98%)	Accuracy	Per Character Accuracy	Per Character Accuracy (Prev. state of the Art)
Public SVHN	95.6%	96%	97.8%	97.7%
Private Dataset	89%	91%		



1180 vs. 1780



1844 vs. 184

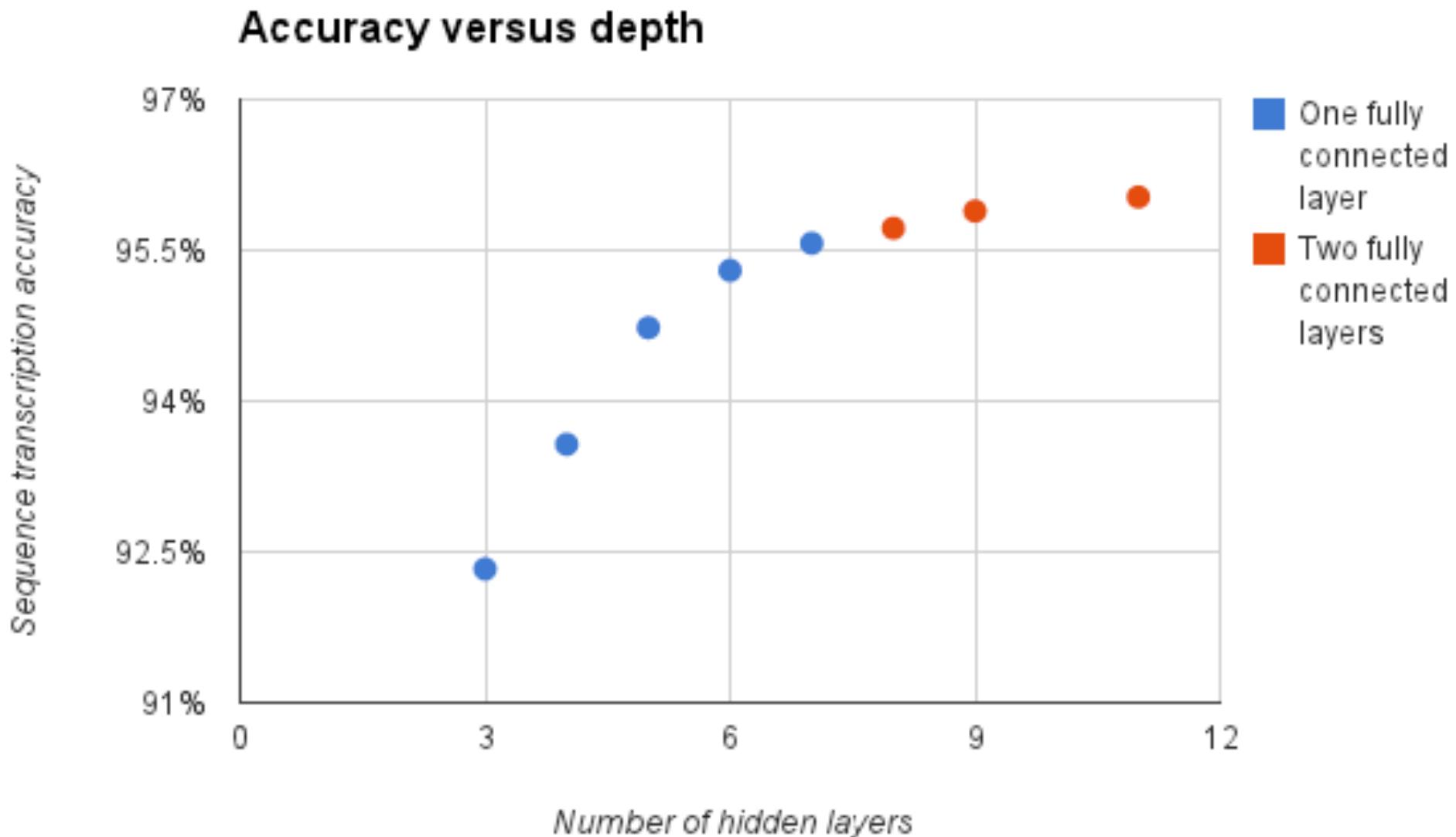


2 vs 239

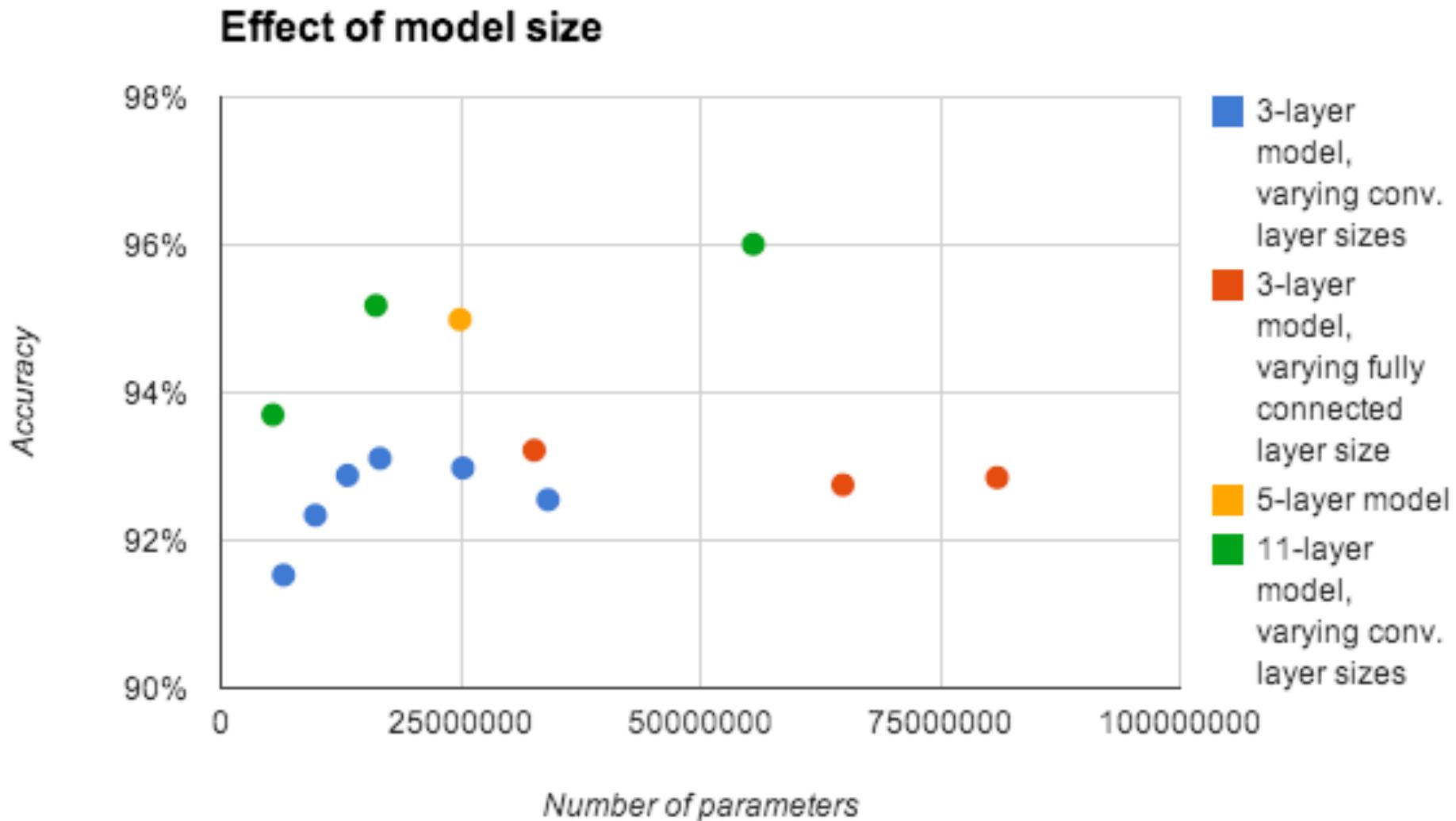


100 vs. 676

## Effect of depth



## Effect of # of parameters



# Conclusion

- Unsupervised learning useful when very little labeled data available
- Generative models useful for missing value problems
- Implicit ensembles and/or lots of data are much more effective