CVPR 2012 Tutorial

Deep Learning Methods for Vision (draft)

Honglak Lee

Computer Science and Engineering Division

University of Michigan, Ann Arbor

Feature representations

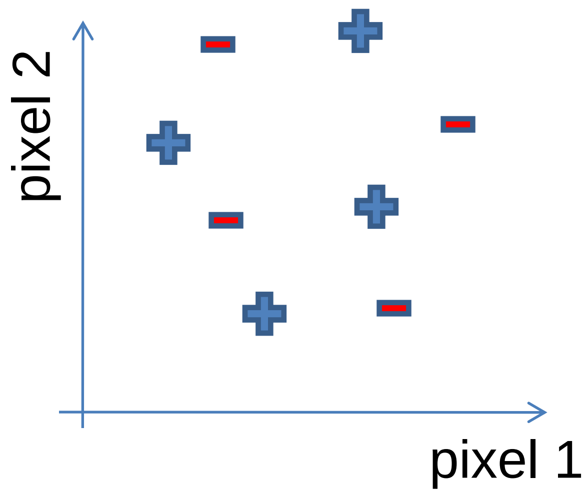
Input

Learning algorithm

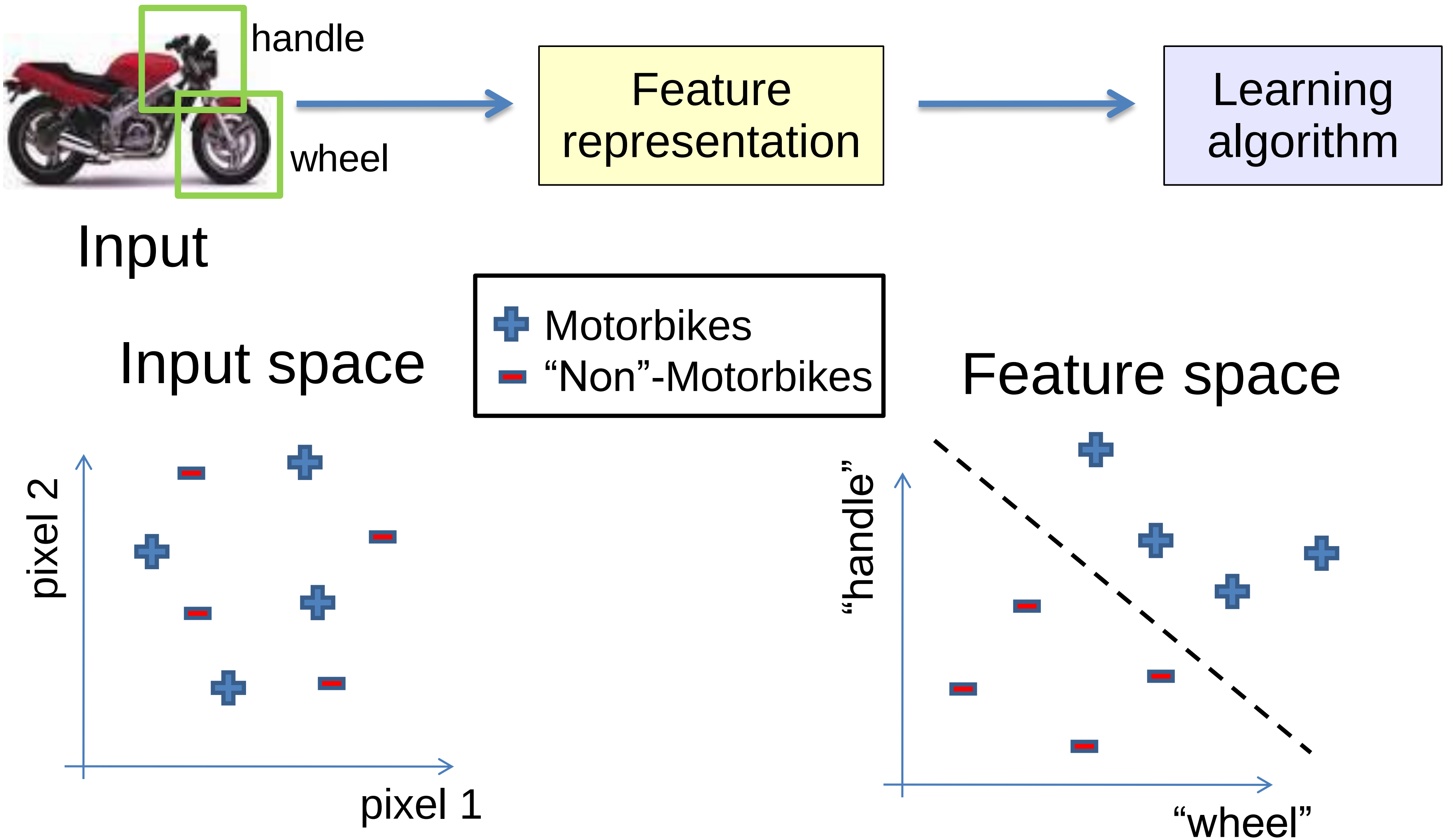
Input space

Motorbikes

"Non"-Motorbikes



Feature representations



How is computer perception done?

State-of-the-art:
“hand-crafting”

Input data → Feature representation → Learning algorithm

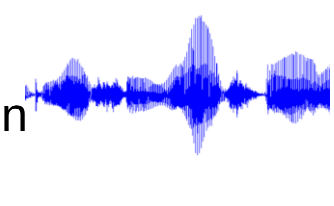
Object detection

Image

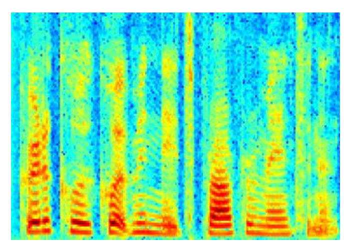
Low-level
vision features
(SIFT, HOG, etc.)

Object detection
/ classification

Audio classification



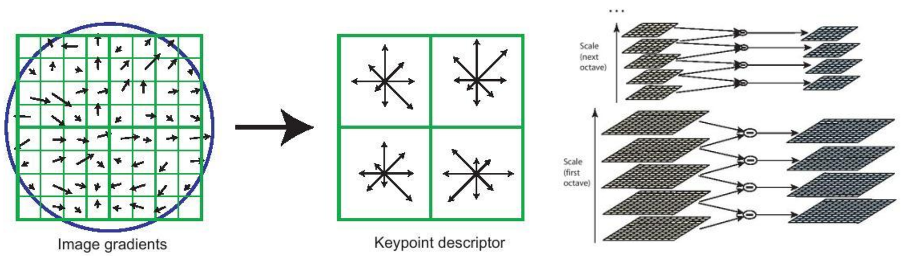
Audio



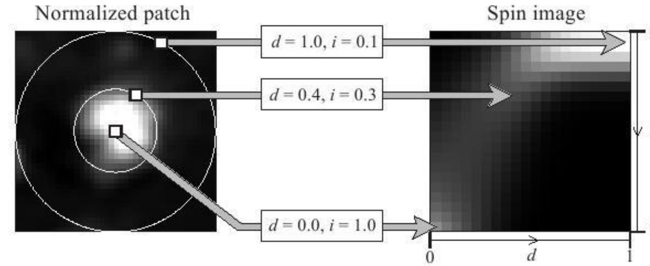
Low-level
audio features
(spectrogram, MFCC, etc.)

Speaker
identification

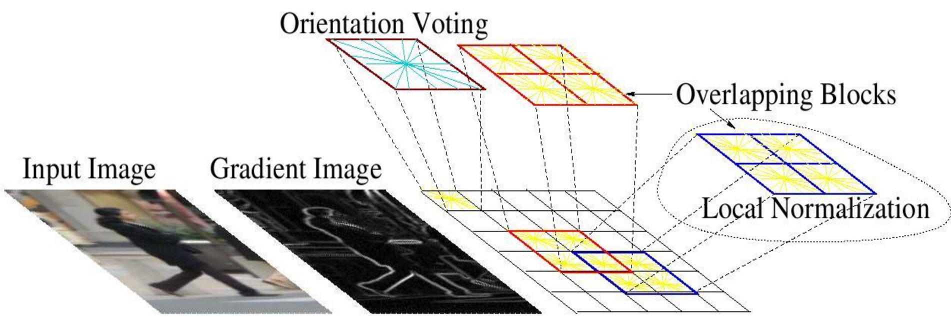
Computer vision features



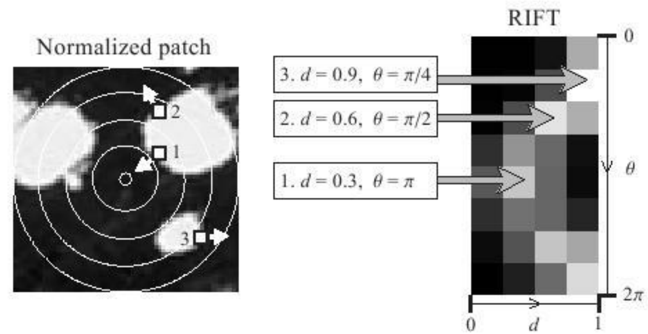
SIFT



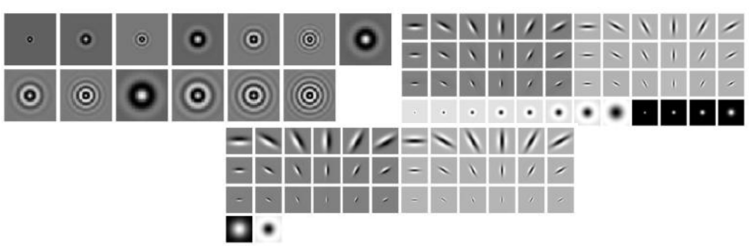
Spin image



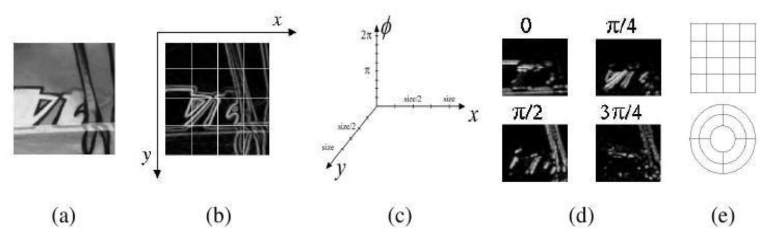
HoG



RIFT

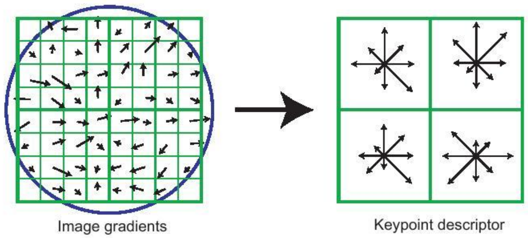


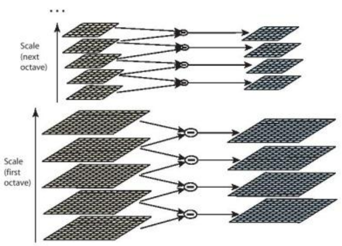
Textons



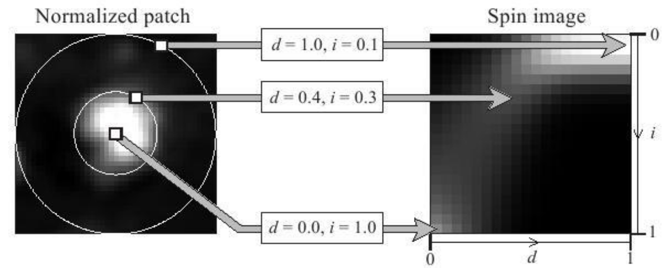
GLOH

Computer vision features

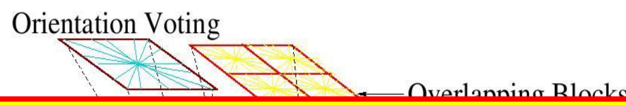


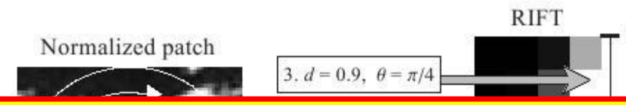


SIFT



Spin image





Hand-crafted features:

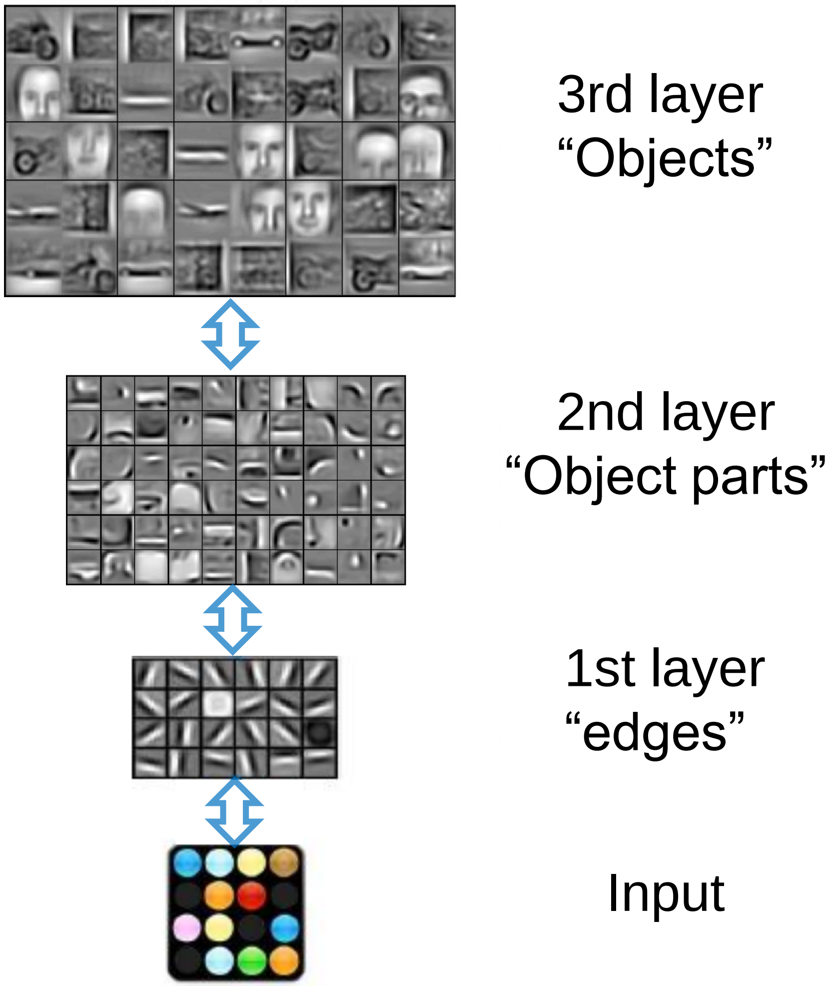
1. Needs expert knowledge
2. Requires time-consuming hand-tuning
3. (Arguably) one of the limiting factors of computer vision systems

Learning Feature Representations

- Key idea:
  - Learn <u>statistical structure or correlation</u> of the data from <u>unlabeled</u> data
  - The learned representations can be used as <u>features</u> in supervised and semi-supervised settings
  - Known as: unsupervised feature learning, feature learning, deep learning, representation learning, etc.
- Topics covered in this talk:
  - Restricted Boltzmann Machines
  - Deep Belief Networks
  - Denoising Autoencoders
  - Applications: Vision, Audio, and Multimodal learning

Learning Feature Hierarchy

- Deep Learning
  - Deep architectures can be representationally efficient.
  - Natural progression from low level to high level structures.
  - Can share the lower-level representations for multiple tasks.

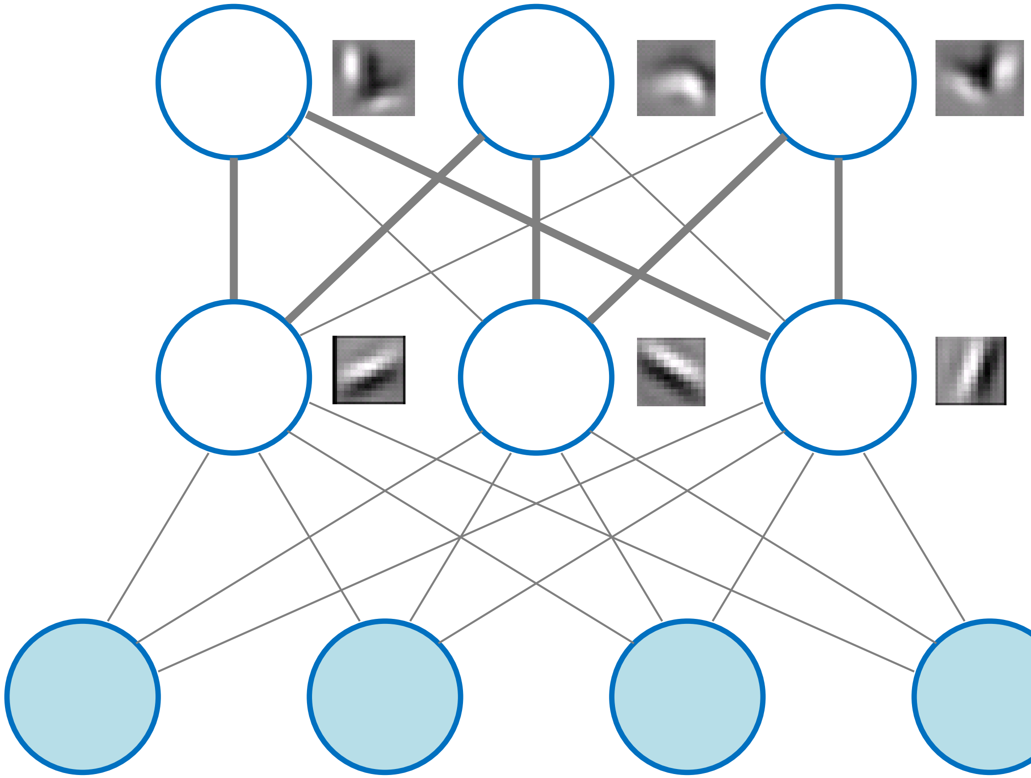


Outline

- Restricted Boltzmann machines
- Deep Belief Networks
- Denoising Autoencoders
- Applications to Vision
- Applications to Audio and Multimodal Data

Learning Feature Hierarchy

[Related work: Hinton, Bengio, LeCun, Ng, and others.]



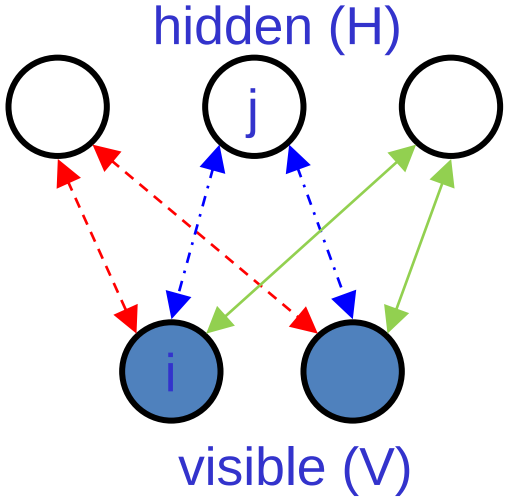
Higher layer: DBNs (Combinations of edges)

First layer: RBMs (edges)

Input image patch (pixels)

Restricted Boltzmann Machines with binary-valued input data

- Representation
  - Undirected bipartite graphical model
  - $\mathbf{v} \in \{0,1\}^D$: observed (visible) binary variables
  - $\mathbf{h} \in \{0,1\}^K$: hidden binary variables.



$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h}))$$

$$E(\mathbf{v},\mathbf{h}) = -\sum_{ij} v_i W_{ij} h_j - \sum_j b_j h_j - \sum_i c_i v_i$$

$$= -\mathbf{v}^T W \mathbf{h} - \mathbf{b}^T \mathbf{h} - \mathbf{c}^T \mathbf{v}$$

$$Z = \sum_{\mathbf{v}\in\{0,1\}^D} \sum_{\mathbf{h}\in\{0,1\}^K} \exp(-E(\mathbf{v},\mathbf{h}))$$

Conditional Probabilities (RBM with binary-valued input data)

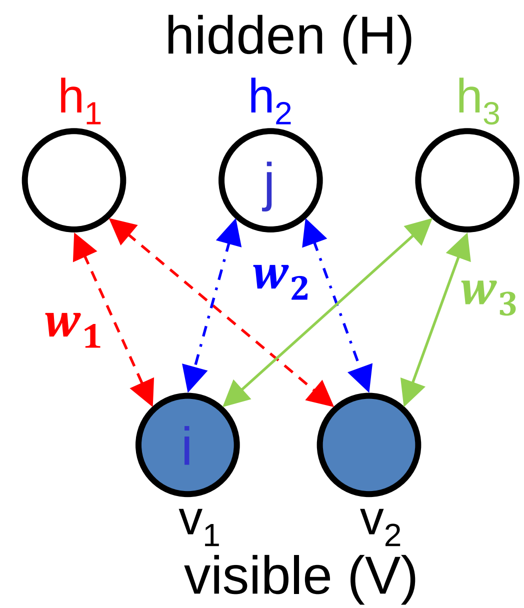
- Given $\mathbf{v}$, all the h_j are conditionally independent

$$P\left(h_j = 1 \middle| \mathbf{v}\right) = \frac{\exp\left(\sum_i W_{ij} v_j + b_j\right)}{\exp\left(\sum_i W_{ij} v_j + b_j\right) + 1}$$

$$= \text{sigmoid}\left(\sum_i W_{ij}\, v_j + b_j\right)$$

$$= \text{sigmoid}\left(\boldsymbol{w}_j^T \mathbf{v} + b_j\right)$$

  - P(**h**|**v**) can be used as “features”

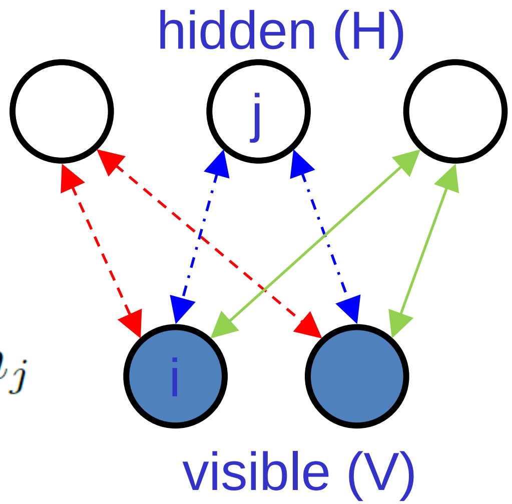


- Given $\mathbf{h}$, all the v_i are conditionally independent

$$P(v_i | \mathbf{h}) = \text{sigmoid}\left(\sum_j W_{ij}\, h_j + c_i\right)$$

Restricted Boltzmann Machines with real-valued input data

- Representation
  - Undirected bipartite graphical model
  - V: observed (visible) real variables
  - H: hidden binary variables.



$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h}))$$

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2} \sum_i (v_i - c_i)^2 - \frac{1}{\sigma} \sum_{i,j} v_i W_{ij} h_j - \sum_j b_j h_j$$

Conditional Probabilities (RBM with real-valued input data)

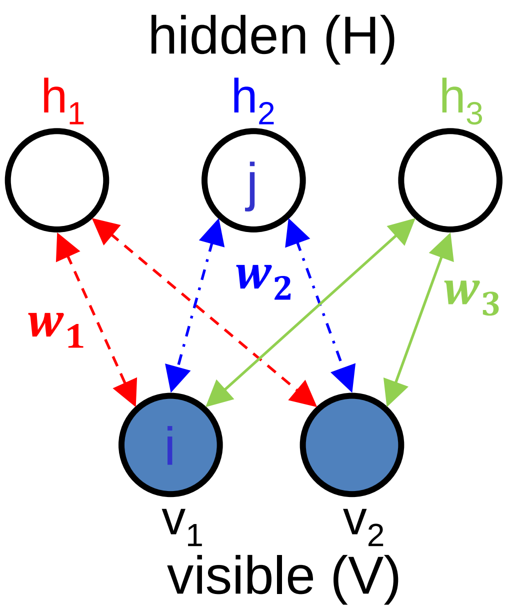
- Given $\mathbf{v}$, all the h_j are conditionally independent

$$P\left(h_j = 1 \middle| \boldsymbol{v}\right) = \frac{\exp\left(\frac{1}{\sigma}\sum_i W_{ij}v_j + b_j\right)}{\exp\left(\frac{1}{\sigma}\sum_i W_{ij}v_j + b_j\right) + 1}$$

$$= \text{sigmoid}\left(\frac{1}{\sigma}\sum_i W_{ij}\, v_j + b_j\right)$$

$$= \text{sigmoid}\left(\frac{1}{\sigma}\boldsymbol{w}_j^T \boldsymbol{v} + b_j\right)$$

  - P(**h**|**v**) can be used as "features"



- Given $\mathbf{h}$, all the v_i are conditionally independent

$P(v_i|\mathbf{h}) = \mathcal{N}\left(\sigma\sum_j W_{ij}\, h_j + c_i, \sigma^2\right)$ or

$P(\mathbf{v}|\mathbf{h}) = \mathcal{N}(\sigma\boldsymbol{W}\mathbf{h} + \mathbf{c}, \sigma^2\mathbf{I})$.

Inference

- Conditional Distribution: P(v|h) or P(h|v)
  - Easy to compute (see previous slides).
  - Due to conditional independence, we can sample all hidden units given all visible units <u>in parallel</u> (and vice versa)
- Joint Distribution: P(v,h)
  - Requires Gibbs Sampling (approximate; lots of iterations to converge).

```
Initialize with v^0
Sample h^0 from P(h|v^0)

Repeat until convergence (t=1,…) {
        Sample v^t from P(v^t|h^(t-1))
        Sample h^t from P(h|v^t)
}
```

Initialize with $\mathbf{v}^0$
Sample $\mathbf{h}^0$ from $P(\mathbf{h}|\mathbf{v}^0)$

Repeat until convergence (t=1,…) {
Sample $\mathbf{v}^t$ from $P(\mathbf{v}^t|\mathbf{h}^{t-1})$
Sample $\mathbf{h}^t$ from $P(\mathbf{h}|\mathbf{v}^t)$
}

Training RBMs

- Model: $P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h};\theta))$
- How can we find parameters θ that maximize $P_{\theta}(\mathbf{v})$?

$$\frac{\partial}{\partial\theta}\log P(\mathbf{v}) = \mathbb{E}_{\mathbf{h}\sim P_\theta(\mathbf{h}|\mathbf{v})}\left[-\frac{\partial}{\partial\theta}E(\mathbf{h},\mathbf{v})\right] - \mathbb{E}_{\mathbf{v}',\mathbf{h}\sim P_\theta(\mathbf{v},\mathbf{h})}\left[-\frac{\partial}{\partial\theta}E(\mathbf{h}',\mathbf{v})\right]$$

Data Distribution
(posterior of h given v)

Model Distribution

- We need to compute P(h|v) and P(v,h), and derivative of E wrt parameters {W,b,c}
  - P(h|v): tractable
  - P(v,h): intractable
    - Can approximate with Gibbs sampling, but requires lots of iterations

Contrastive Divergence

- An approximation of the log-likelihood gradient for RBMs
  1. Replace the average over all possible inputs by samples

$$\frac{\partial}{\partial \theta} \log P(\mathbf{v}) = \mathbb{E}_{\mathbf{h} \sim P_\theta(\mathbf{h}|\mathbf{v})} \left[-\frac{\partial}{\partial \theta} E(\mathbf{h}, \mathbf{v}) \right] - \mathbb{E}_{\mathbf{v}', \mathbf{h} \sim P_\theta(\mathbf{v}, \mathbf{h})} \left[-\frac{\partial}{\partial \theta} E(\mathbf{h}', \mathbf{v}) \right]$$

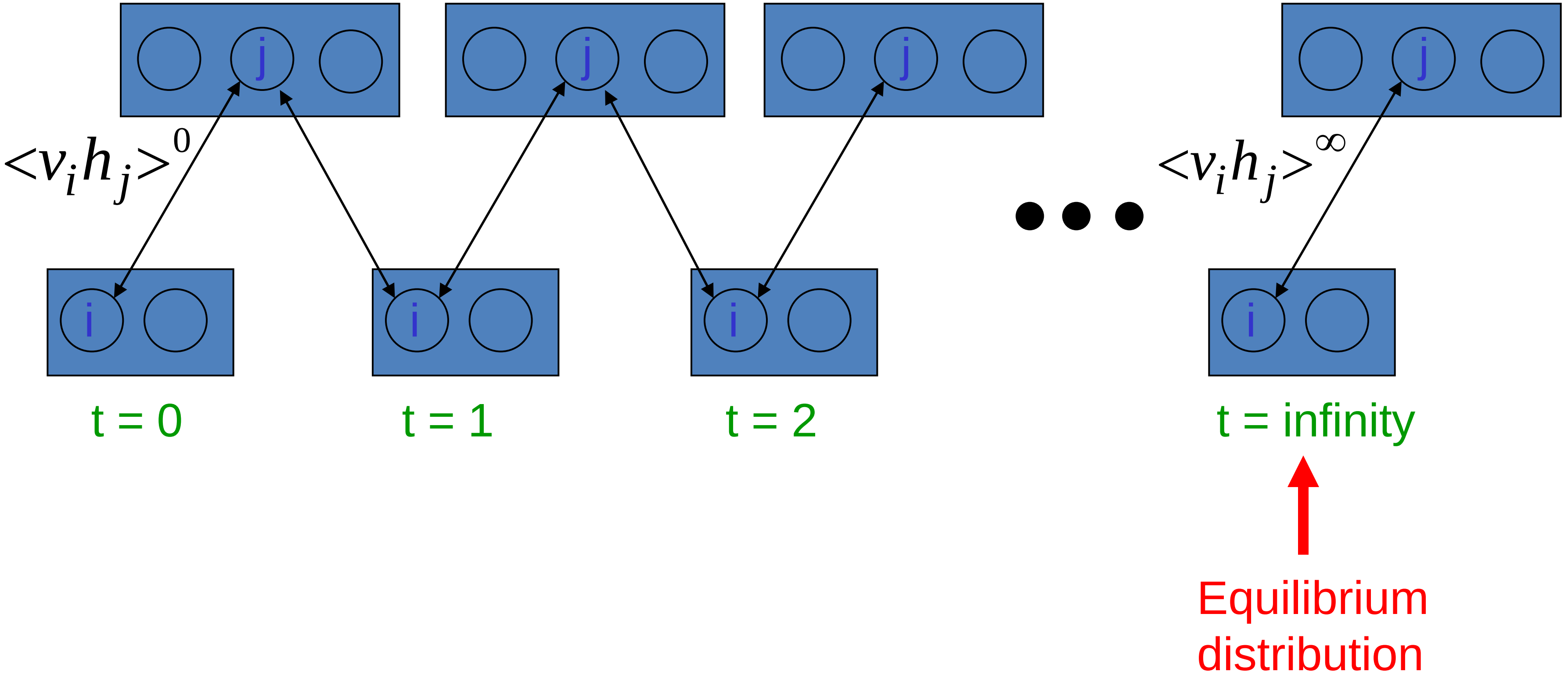
  2. Run the MCMC chain (Gibbs sampling) for only k steps starting from the observed example

```
Initialize with v^0 = v
Sample h^0 from P(h|v^0)

For t = 1,…,k {
        Sample v^t from P(v^t|h^{t-1})
        Sample h^t from P(h|v^t)
}
```

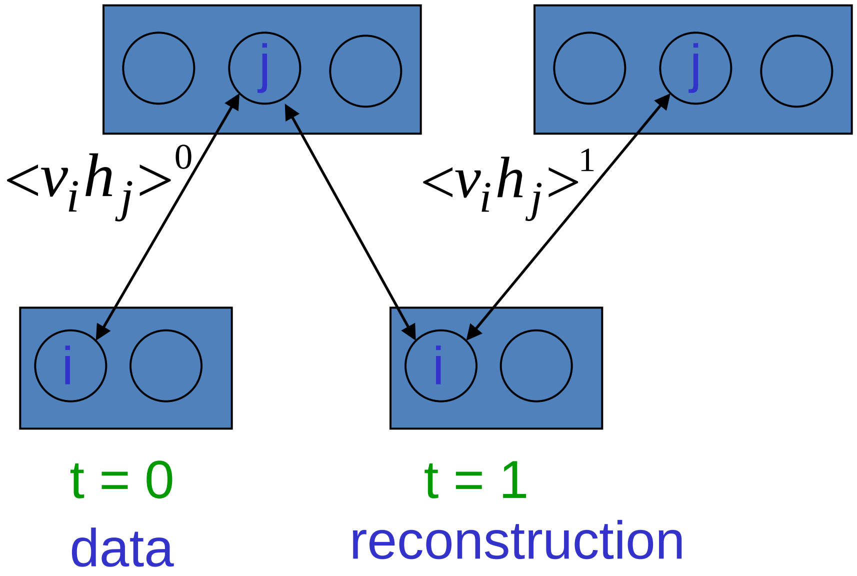

18

A picture of the maximum likelihood learning algorithm for an RBM



Slide Credit: Geoff Hinton

A quick way to learn an RBM



Start with a training vector on the visible units.

Update all the hidden units in parallel

Update the all the visible units in parallel to get a "reconstruction".

Update the hidden units again.

Slide Credit: Geoff Hinton

Update Rule: Putting together

- Training via stochastic gradient.
- Note, $\frac{\partial E}{\partial W_{ij}} = h_i v_j$.
- Therefore,

$$\frac{\partial}{\partial W_{ij}} \log P(\mathbf{v}) = \mathbb{E}_{\mathbf{h} \sim P_\theta(\mathbf{h}|\mathbf{v})} [v_i h_j] - \mathbb{E}_{\mathbf{v}', \mathbf{h} \sim P_\theta(\mathbf{v}, \mathbf{h})} [v_i h_j]$$

  – Can derive similar update rule for biases *b* and *c*
  – Implemented in ~10 lines of matlab code

Other ways of training RBMs

- Persistent CD [Tieleman, ICML 2008; Tieleman & Hinton, ICML 2009]
  - Keep a background MCMC chain to obtain the negative phase samples.
  - Related to Stochastic Approximation
    - Robbins and Monro, Ann. Math. Stats, 1957
    - L. Younes, Probability Theory 1989
- Score Matching [Swersky et al., ICML 2011; Hyvarinen, JMLR 2005]
  - Use score function to eliminate Z
  - Match model's & empirical score function

$$p(x) = q(x)/Z \qquad \psi = \frac{\partial \log p(x)}{\partial x} = \frac{\partial \log q(x)}{\partial x}$$

Estimating Log-Likelihood of RBMs

- How do we measure the likelihood of the learned models?
- RBM: requires estimating partition function
  - Reconstruction error provides a cheap proxy
  - Log Z tractable analytically for < 25 binary inputs or hidden
  - Can be approximated with Annealed Importance Sampling (AIS)
    - Salakhutdinov & Murray, ICML 2008
- Open question: efficient ways to monitor progress

Variants of RBMs

Sparse RBM / DBN

- Main idea [Lee et al., NIPS 2008]
  - Constrain the hidden layer nodes to have "sparse" average values (activation). [cf. sparse coding]
- Optimization
  - Tradeoff between "likelihood" and "sparsity penalty"

$$\text{minimize}_{\{W,b,c\}} \underbrace{- \sum_{k=1}^{m} \log P(\mathbf{v}^{(k)})}_{\text{Log-likelihood}} + \underbrace{\phi(W,b,c)}_{\text{Sparsity penalty}}$$

where $\phi \triangleq \lambda \sum_j (\mathbb{E}_{sample}[h_j|\mathbf{v}] - p)^2$

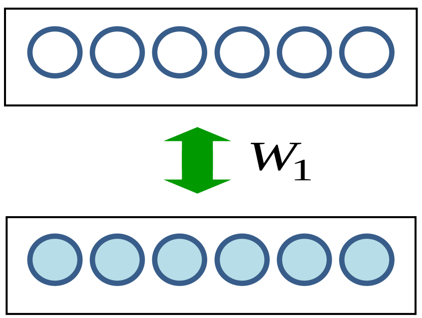
Average activation

Target sparsity

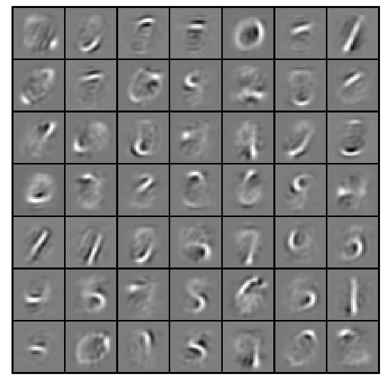
we can use other penalty functions (e.g., KLdivergence)

Modeling handwritten digits

- Sparse dictionary learning via sparse RBMs



input nodes (data)



First layer bases
("pen-strokes")

Training examples

Learned sparse representations can be used as features.

[Lee et al., NIPS 2008; Ranzato et al, NIPS 2007]

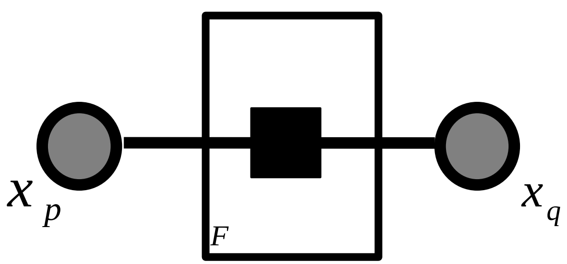
3-way factorized RBM

[Ranzato et al., AISTATS 2010; Ranzato and Hinton, CVPR 2010]

- Models the covariance structure of images using hidden variables
  - 3-way factorized RBM / mean-covariance RBM

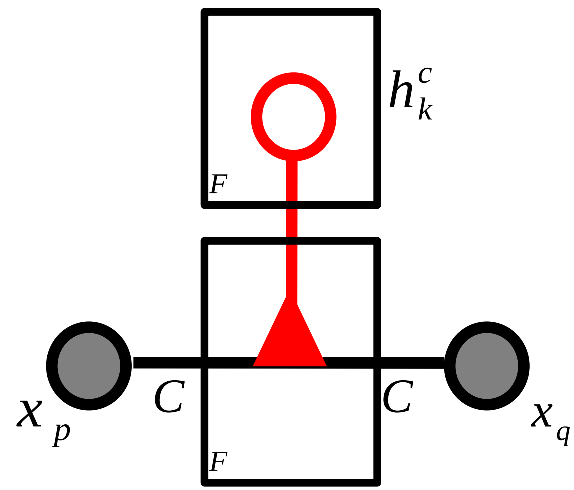
Gaussian MRF

$$E(\mathbf{x},\mathbf{h}) = \frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x}$$



3-way (covariance) RBM

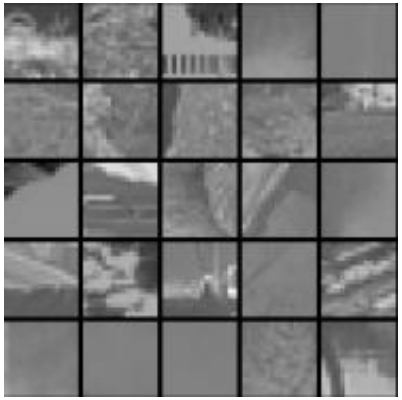
$$E(\mathbf{x},\mathbf{h}) = \frac{1}{2}\mathbf{x}^T C[diag(P\mathbf{h})]C^T \mathbf{x}$$

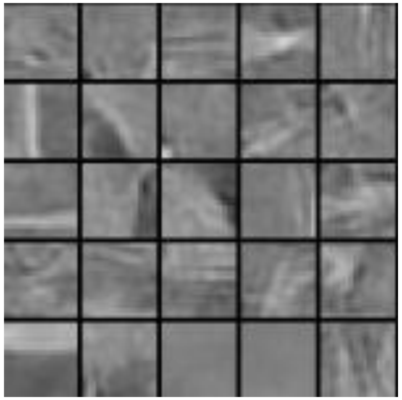


[Slide Credit: Marc'Aurelio Ranzato]

Generating natural image patches

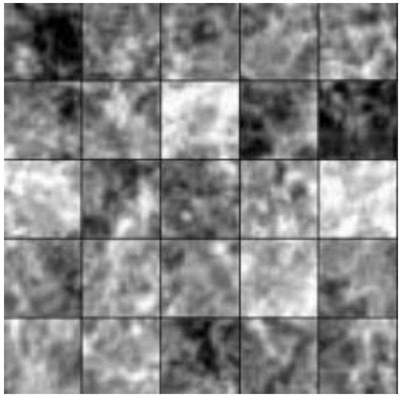
Natural images





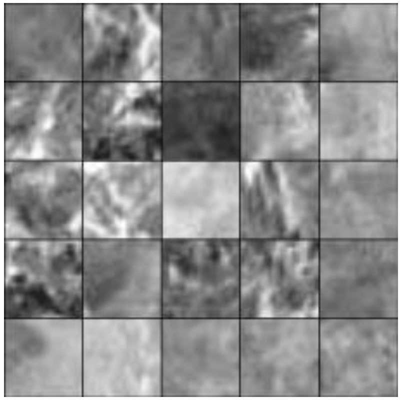
mcRBM

Ranzato and Hinton CVPR 2010



GRBM

from Osindero and Hinton NIPS 2008



S-RBM + DBN

from Osindero and Hinton NIPS 2008

Slide Credit: Marc'Aurelio Ranzato

Outline

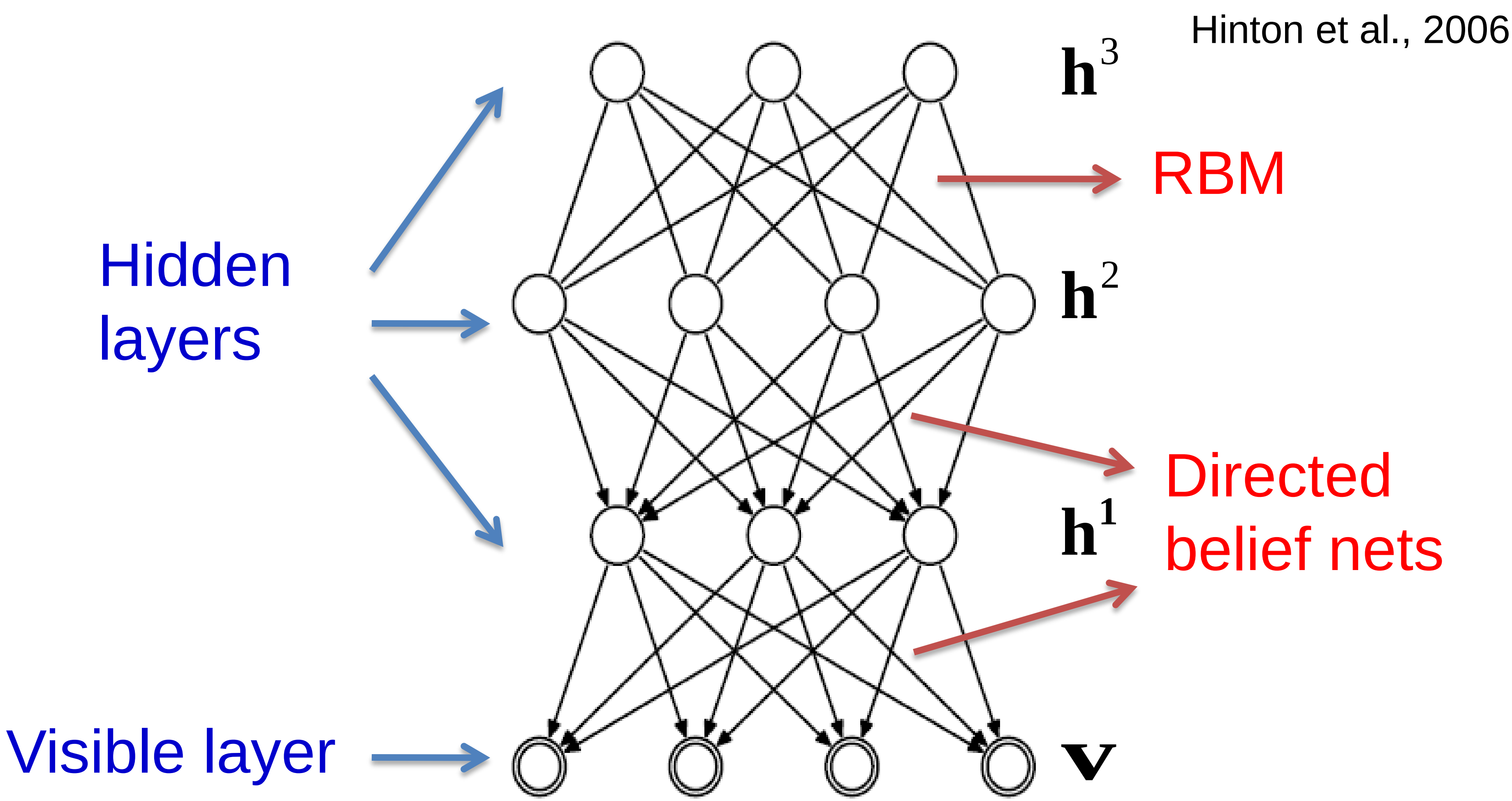
- Restricted Boltzmann machines
- Deep Belief Networks
- Denoising Autoencoders
- Applications to Vision
- Applications to Audio and Multimodal Data

Deep Belief Networks (DBNs)

Hinton et al., 2006

- Probabilistic generative model
- Deep architecture – multiple layers
- Unsupervised pre-learning provides a good initialization of the network
  - maximizing the lower-bound of the log-likelihood of the data
- Supervised fine-tuning
  - Generative: Up-down algorithm
  - Discriminative: backpropagation

DBN structure



$$P(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^l) = P(\mathbf{v}\,|\,\mathbf{h}^1)P(\mathbf{h}^1\,|\,\mathbf{h}^2)\ldots P(\mathbf{h}^{l-2}\,|\,\mathbf{h}^{l-1})P(\mathbf{h}^{l-1},\mathbf{h}^l)$$

DBN structure

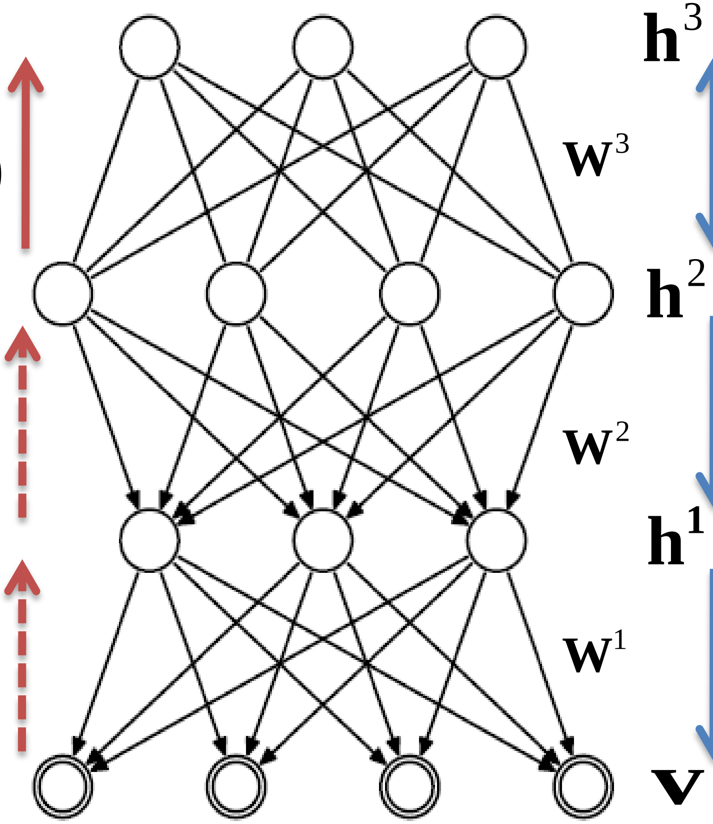
Hinton et al., 2006

(approximate) inference

$Q(\mathbf{h}^3 \mid \mathbf{h}^2) = P(\mathbf{h}^3 \mid \mathbf{h}^2)$

$Q(\mathbf{h}^2 \mid \mathbf{h}^1)$

$Q(\mathbf{h}^1 \mid \mathbf{v})$



Generative process

$P(\mathbf{h}^2, \mathbf{h}^3)$

$P(\mathbf{h}^1 \mid \mathbf{h}^2)$

$P(\mathbf{v} \mid \mathbf{h}^1)$

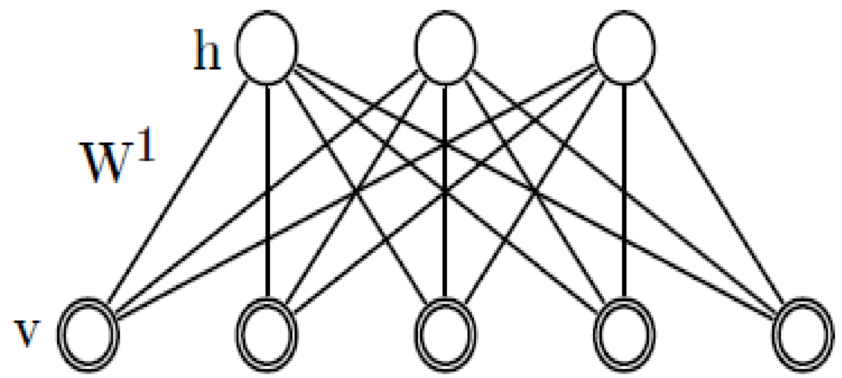
$$P(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^l) = P(\mathbf{v} \mid \mathbf{h}^1) P(\mathbf{h}^1 \mid \mathbf{h}^2) \ldots P(\mathbf{h}^{l-2} \mid \mathbf{h}^{l-1}) P(\mathbf{h}^{l-1}, \mathbf{h}^l)$$

$$Q(\mathbf{h}^i \mid \mathbf{h}^{i-1}) = \prod_j sigm(\mathbf{b}_j^{i-1} + \mathrm{W}_j^i \mathbf{h}^{i-1}) \quad P(\mathbf{h}^{i-1} \mid \mathbf{h}^i) = \prod_j sigm(\mathbf{b}_j^i + \mathrm{W}_{.j}^i{}' \mathbf{h}^i)$$

DBN Greedy training

Hinton et al., 2006

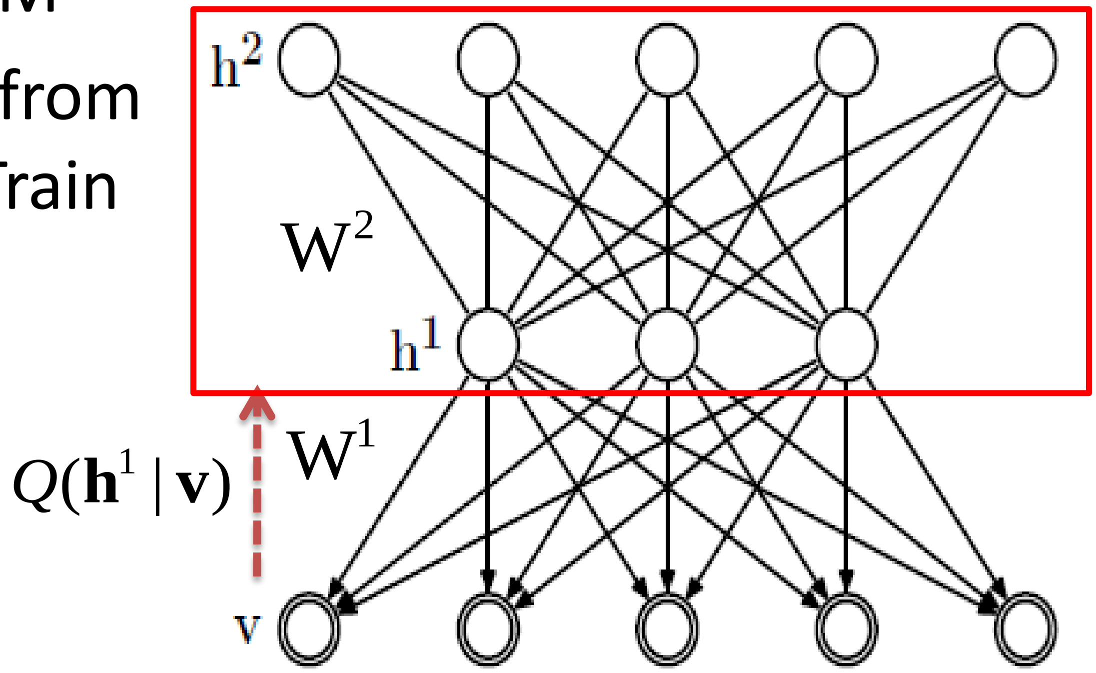
- First step:
  - Construct an RBM with an input layer **v** and a hidden layer **h**
  - Train the RBM



DBN Greedy training

Hinton et al., 2006

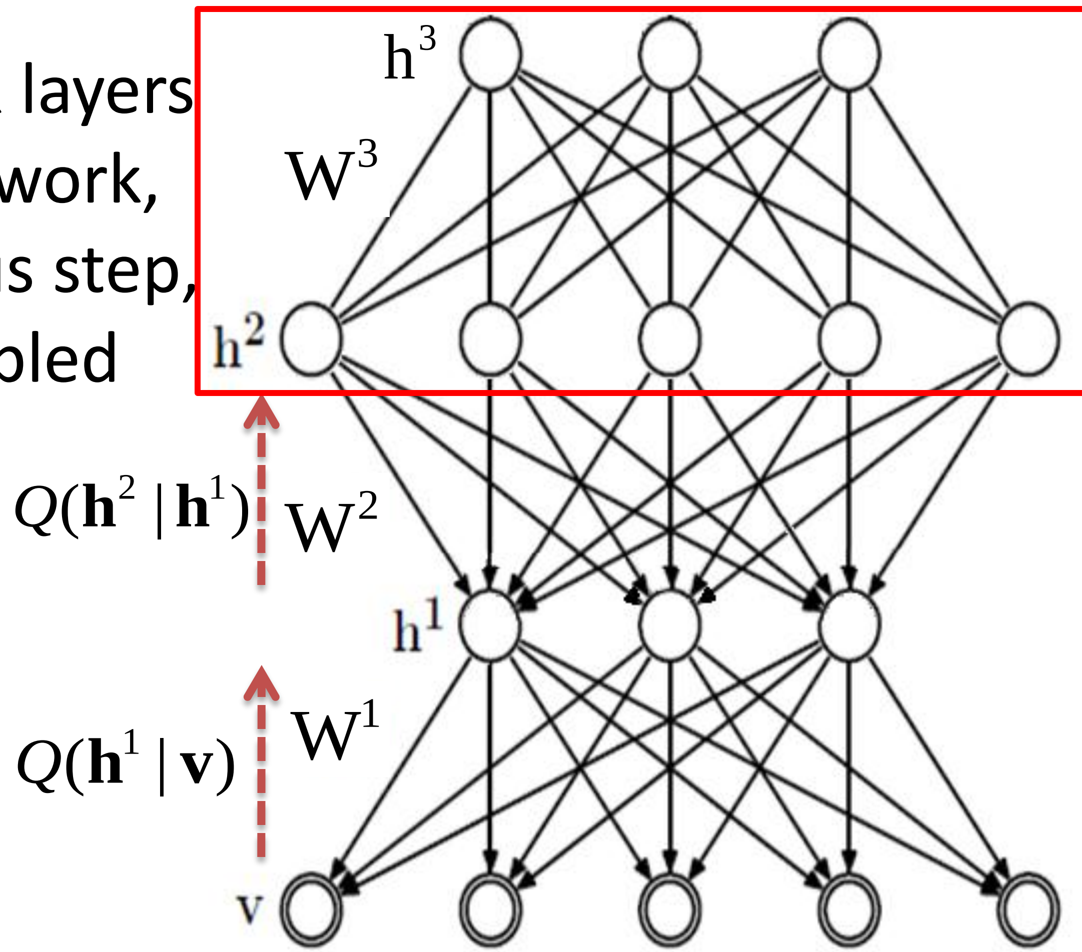
- Second step:
  - Stack another hidden layer on top of the RBM to form a new RBM
  - Fix W^1, sample $\mathbf{h}^1$ from $Q(\mathbf{h}^1 \mid \mathbf{v})$ as input. Train W^2 as RBM.



DBN Greedy training

Hinton et al., 2006

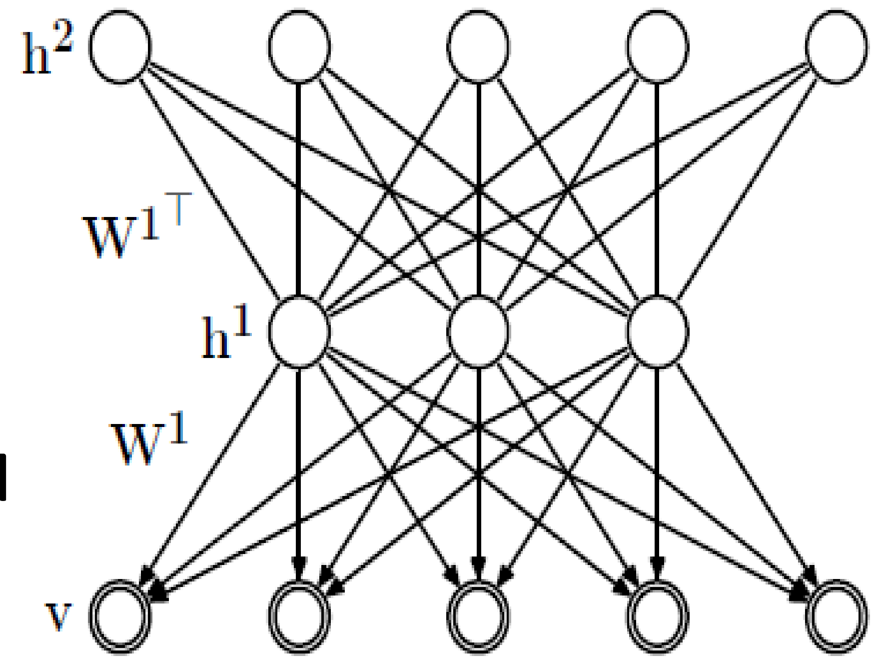
- Third step:
  – Continue to stack layers on top of the network, train it as previous step, with sample sampled from $Q(\mathbf{h}^2 \mid \mathbf{h}^1)$
- And so on…



Why greedy training works?

Hinton et al., 2006

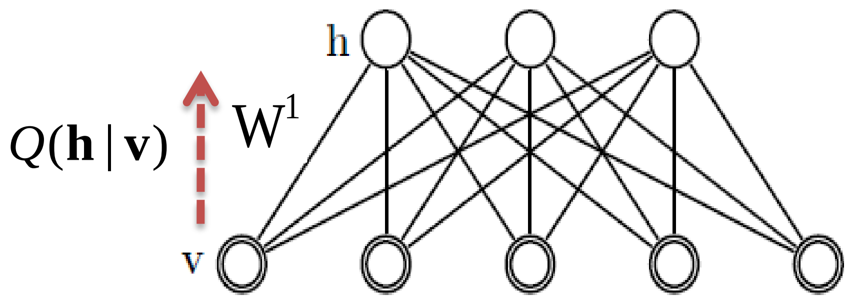
- RBM specifies P(v,h) from P(v|h) and P(h|v)
  - Implicitly defines P(v) and P(h)
- Key idea of stacking
  - Keep P(v|h) from 1st RBM
  - Replace P(h) by the distribution generated by 2nd level RBM



Why greedy training works?

Hinton et al., 2006

- Greey Training:
  - Variational lower-bound justifies greedy layerwise training of RBMs



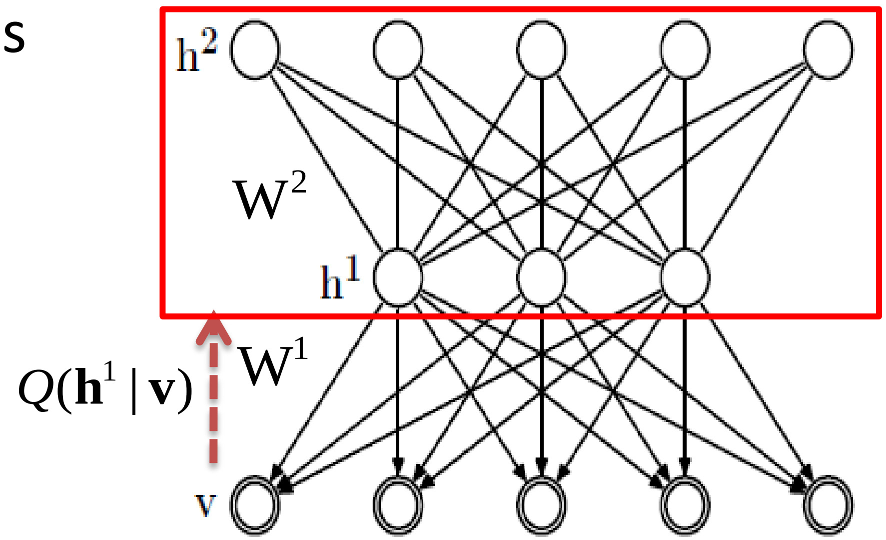
$$\log P(\mathbf{x}) \geq H_{Q(\mathbf{h}|\mathbf{x})} + \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{x}) \left(\underline{\log P(\mathbf{h})} + \log P(\mathbf{x}|\mathbf{h})\right)$$

Trained by the second layer RBM

Why greedy training works?

Hinton et al., 2006

- Greey Training:
  - Variational lower-bound justifies greedy layerwise training of RBMs
  - Note: RBM and 2-layer DBN are equivalent when $W^2 = (W^1)^T$. Therefore, the lower bound is tight and the log-likelihood improves by greedy training.



$$\log P(\mathbf{x}) \geq H_{Q(\mathbf{h}|\mathbf{x})} + \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{x}) \left(\underline{\log P(\mathbf{h})} + \log P(\mathbf{x}|\mathbf{h})\right)$$

Trained by the second layer RBM

DBN and supervised fine-tuning

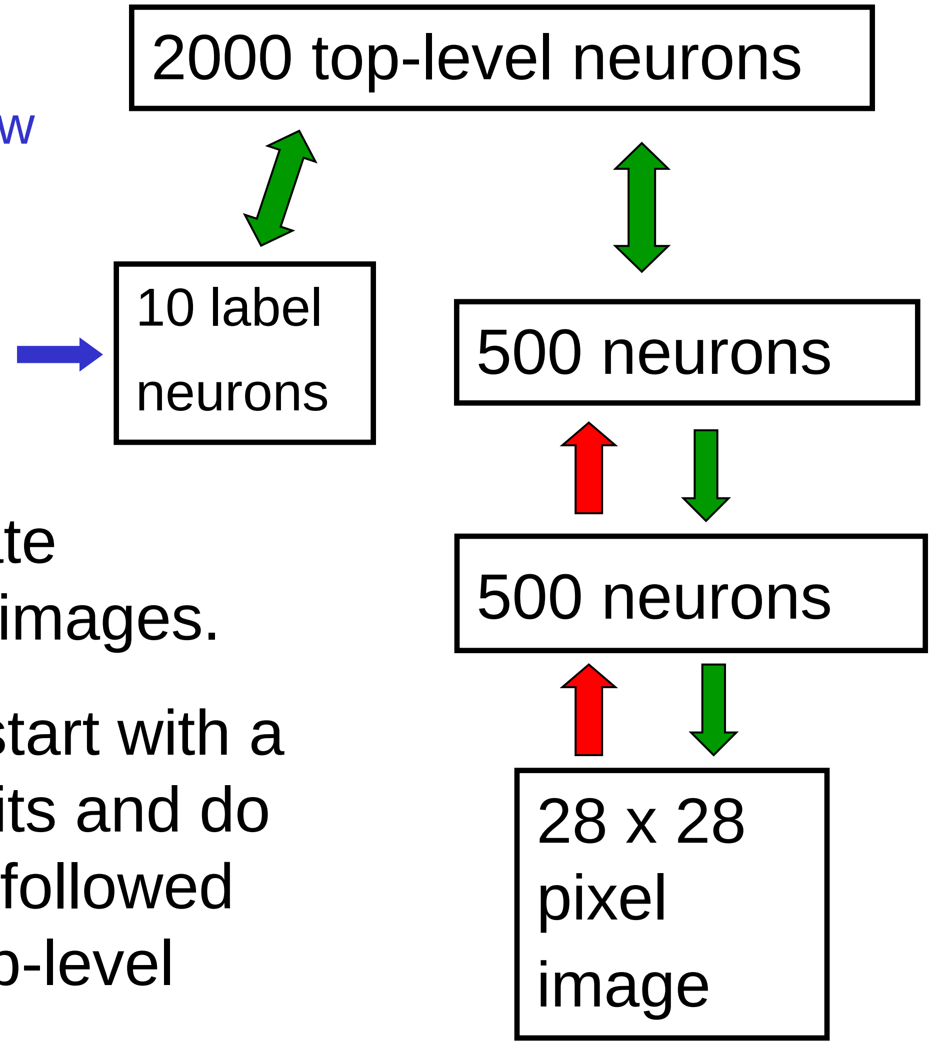
- Discriminative fine-tuning
  - Initializing with neural nets + backpropagation
  - Maximizes $\log P(Y \mid X)$ (X: data Y: label)

- Generative fine-tuning
  - Up-down algorithm
  - Maximizes $\log P(Y, X)$ (joint likelihood of data and labels)

A model for digit recognition

The top two layers form an associative memory whose energy landscape models the low dimensional manifolds of the digits.

The energy valleys have names



The model learns to generate combinations of labels and images.

To perform recognition we start with a neutral state of the label units and do an up-pass from the image followed by a few iterations of the top-level associative memory.

http://www.cs.toronto.edu/~hinton/nipstutorial/nipstut3.ppt

Slide Credit: Geoff Hinton

Fine-tuning with a contrastive version of the “wake-sleep” algorithm

After learning many layers of features, we can fine-tune the features to improve generation.

1. Do a stochastic bottom-up pass
   – Adjust the top-down weights to be good at reconstructing the feature activities in the layer below.
2. Do a few iterations of sampling in the top level RBM
   -- Adjust the weights in the top-level RBM.
3. Do a stochastic top-down pass
   – Adjust the bottom-up weights to be good at reconstructing the feature activities in the layer above.

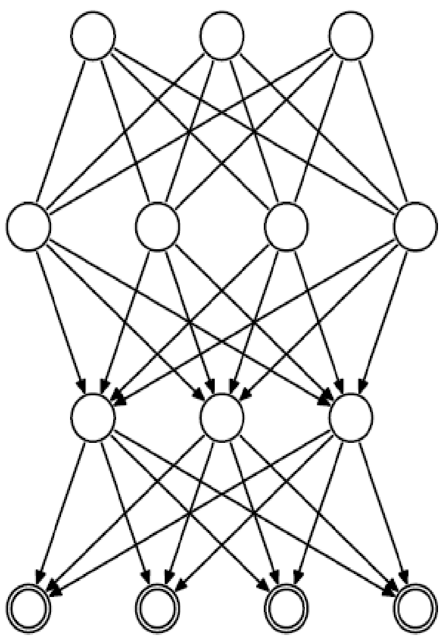
http://www.cs.toronto.edu/~hinton/nipstutorial/nipstut3.ppt Slide Credit: Geoff Hinton

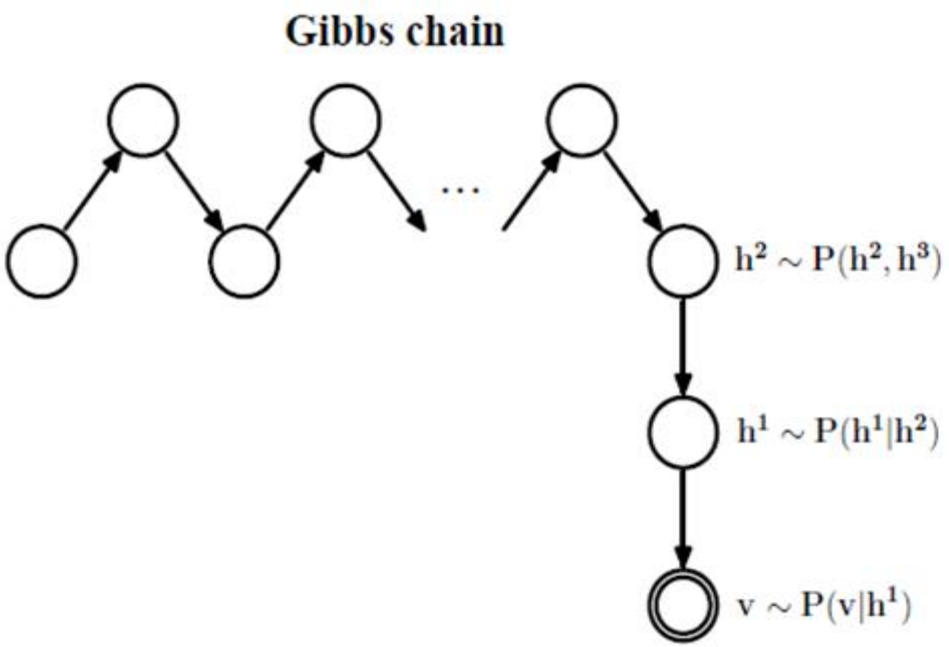
Generating sample from a DBN

- Want to sample from

$$P(\mathbf{v}, \mathbf{h^1}, \mathbf{h^2}, \ldots, \mathbf{h}^l) = P(\mathbf{v} \mid \mathbf{h^1}) P(\mathbf{h^1} \mid \mathbf{h^2}) \ldots P(\mathbf{h}^{l-2} \mid \mathbf{h}^{l-1}) P(\mathbf{h}^{l-1}, \mathbf{h}^l)$$

  – Sample $\mathbf{h}^{l-1}$ using Gibbs sampling in the RBM
  – Sample the lower layer $\mathbf{h}^{i-1}$ from $P(\mathbf{h}^{i-1} \mid \mathbf{h}^i)$





Generating samples from DBN

Figure 9: Each row shows 10 samples from the generative model with a particular label clamped on. The top-level associative memory is initialized by an up-pass from a random binary image in which each pixel is on with a probability of 0.5. The first column shows the results of a down-pass from this initial high-level state. Subsequent columns are produced by 20 iterations of alternating Gibbs sampling in the associative memory.

Hinton et al, A Fast Learning Algorithm for Deep Belief Nets, 2006

Result for supervised fine-tuning on MNIST

• Very carefully trained backprop net with one or two hidden layers (Platt; Hinton)	1.6%
• SVM (Decoste & Schoelkopf, 2002)	1.4%
• Generative model of joint density of images and labels (+ generative fine-tuning)	1.25%
• Generative model of unlabelled digits followed by gentle backpropagation (Hinton & Salakhutdinov, Science 2006)	1.15%

http://www.cs.toronto.edu/~hinton/nipstutorial/nipstut3.ppt

Slide Credit: Geoff Hinton

- More details on up-down algorithm:
  - Hinton, G. E., Osindero, S. and Teh, Y. (2006) “A fast learning algorithm for deep belief nets”, Neural Computation, 18, pp 1527-1554. http://www.cs.toronto.edu/~hinton/absps/ncfast.pdf

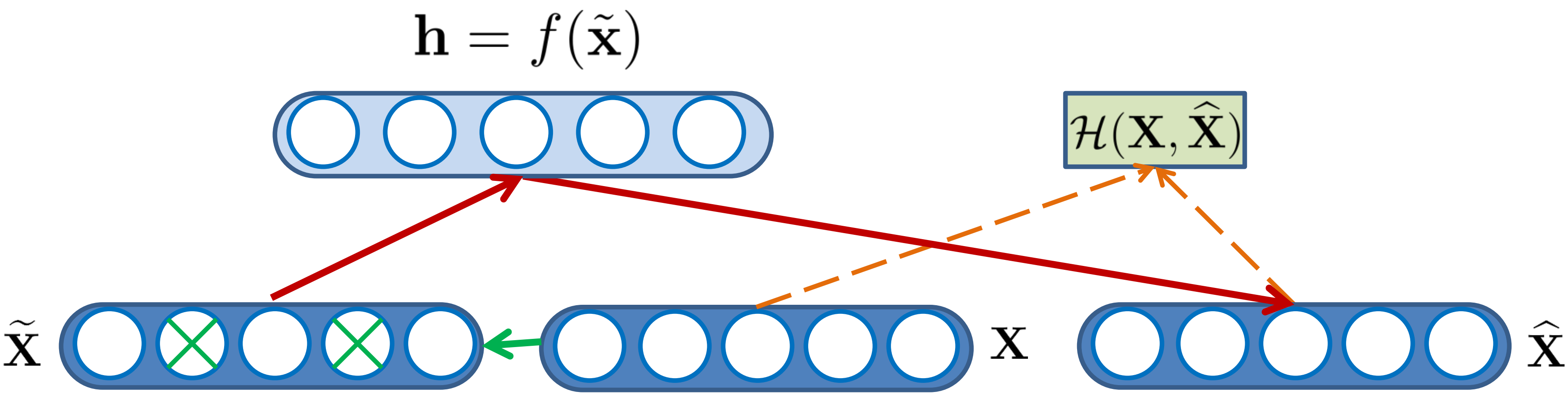
- Handwritten digit demo:
  - http://www.cs.toronto.edu/~hinton/digits.html

Outline

- Restricted Boltzmann machines
- Deep Belief Networks
- Denoising Autoencoders
- Applications to Vision
- Applications to Audio and Multimodal Data

Denoising Autoencoder

- Denoising Autoencoder
  - Perturbs the input **x** to a corrupted version: $\widetilde{\mathbf{x}} \sim q(\widetilde{\mathbf{x}}|\mathbf{x})$
    - E.g., randomly sets some of the coordinates of input to zeros.
  - Recover **x** from encoded **h** of perturbed data.
  - Minimize loss between **x** and **x̂**

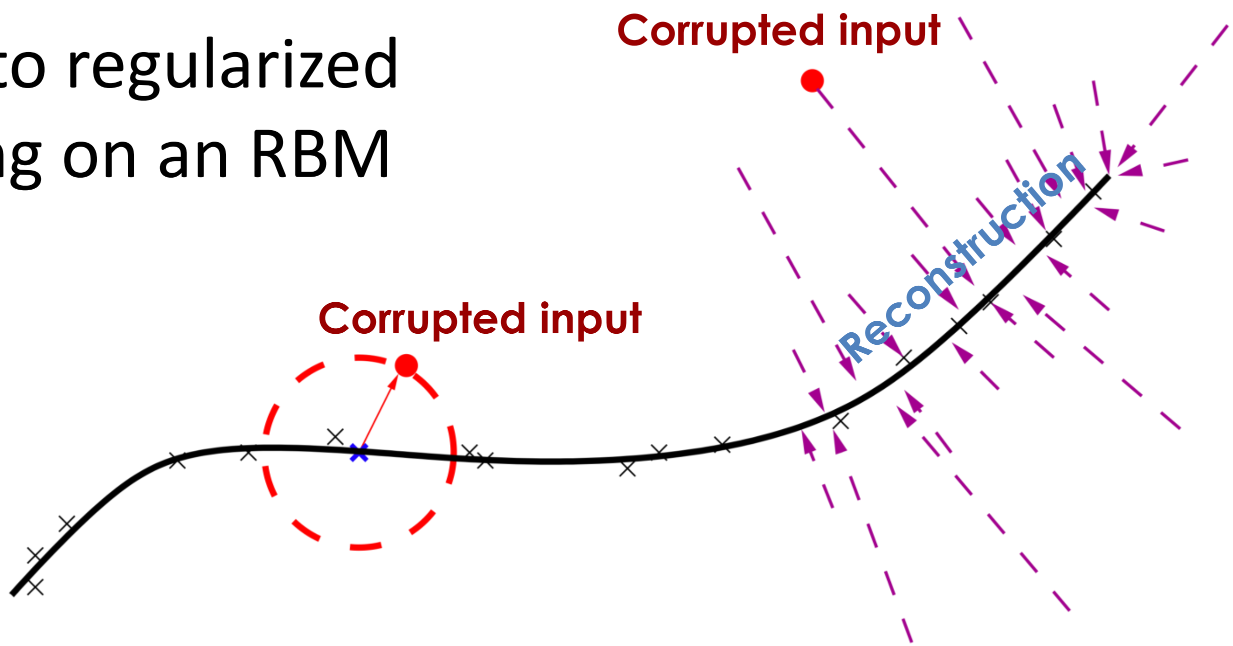


[Vincent et al., ICML 2008]

Denoising Autoencoder

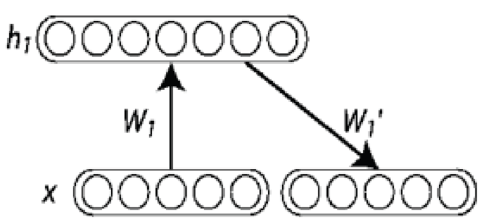
[Vincent et al., ICML 2008]

- Learns a vector field towards higher probability regions
- Minimizes variational lower bound on a generative model
- Corresponds to regularized score matching on an RBM



Stacked Denoising Autoencoders

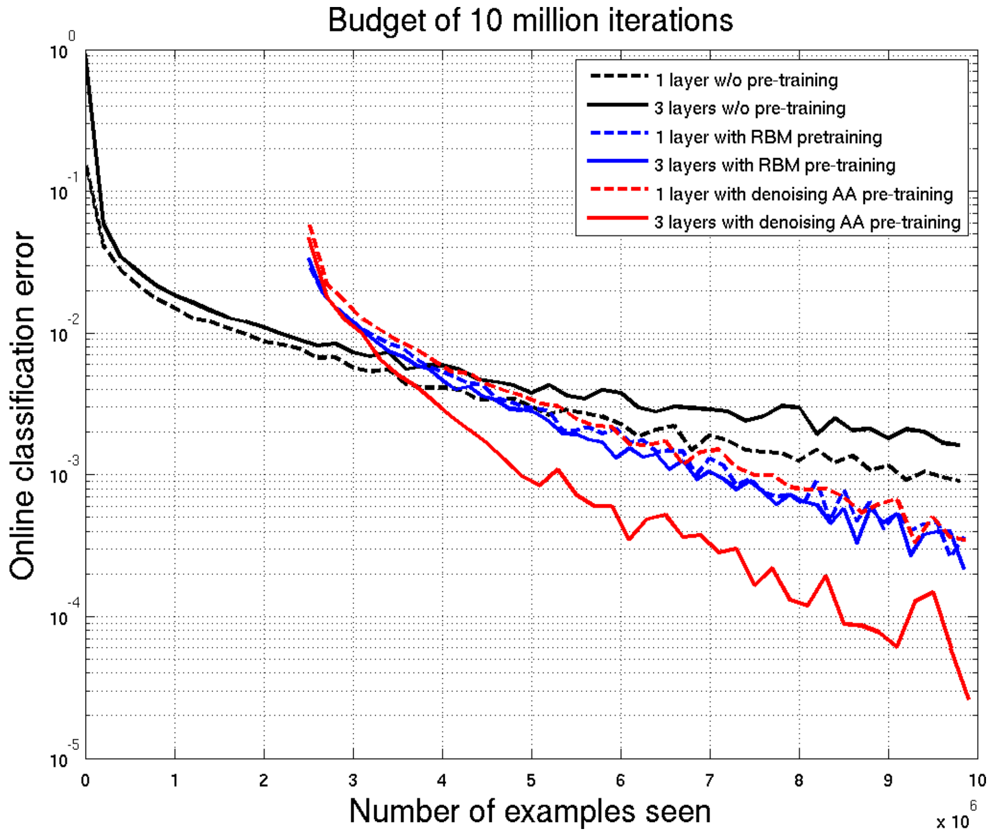
- Greedy Layer wise learning
  - Start with the lowest level and stack upwards
  - Train each layer of autoencoder on the intermediate code (features) from the layer below
  - Top layer can have a different output (e.g., softmax non-linearity) to provide an output for classification



Slide Credit: Yoshua Bengio

Stacked Denoising Autoencoders

- No partition function, can measure training criterion
- Encoder & decoder: any parametrization
- Performs as well or better than stacking RBMs for usupervised pre-training

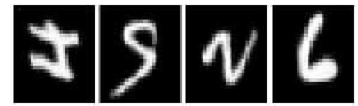
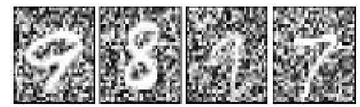
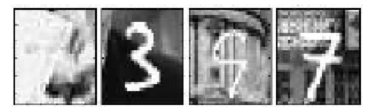
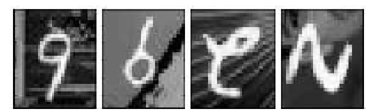
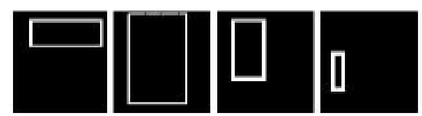
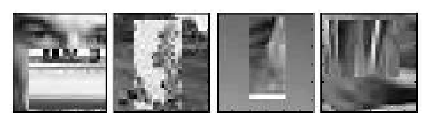


Infinite MNIST

Slide Credit: Yoshua Bengio

Denoising Autoencoders: Benchmarks

Larochelle et al., 2009

basic: subset of MNIST digits.	(10 000 training samples)
rot: applied random rotation (angle between 0 and 2π radians)	
bg-rand: background made of random pixels (value in $0 \ldots 255$)	
bg-img: background is random patch from one of 20 images	
rot-bg-img: combination of rotation and background image	
rect: discriminate between tall and wide rectangles.	
rect-img: same but rectangles are random image patches	
convex: discriminate between convex and non-convex shapes.	

Denoising Autoencoders: Results

- Test errors on the benchmarks

Larochelle et al., 2009

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

Slide Credit: Yoshua Bengio

Why Greedy Layer Wise Training Works

(Bengio 2009, Erhan et al. 2009)

- Regularization Hypothesis
  - Pre-training is "constraining" parameters in a region relevant to unsupervised dataset
  - Better generalization

    (Representations that better describe unlabeled data are more discriminative for labeled data)

- Optimization Hypothesis
  - Unsupervised training initializes lower level parameters near localities of better minima than random initialization can

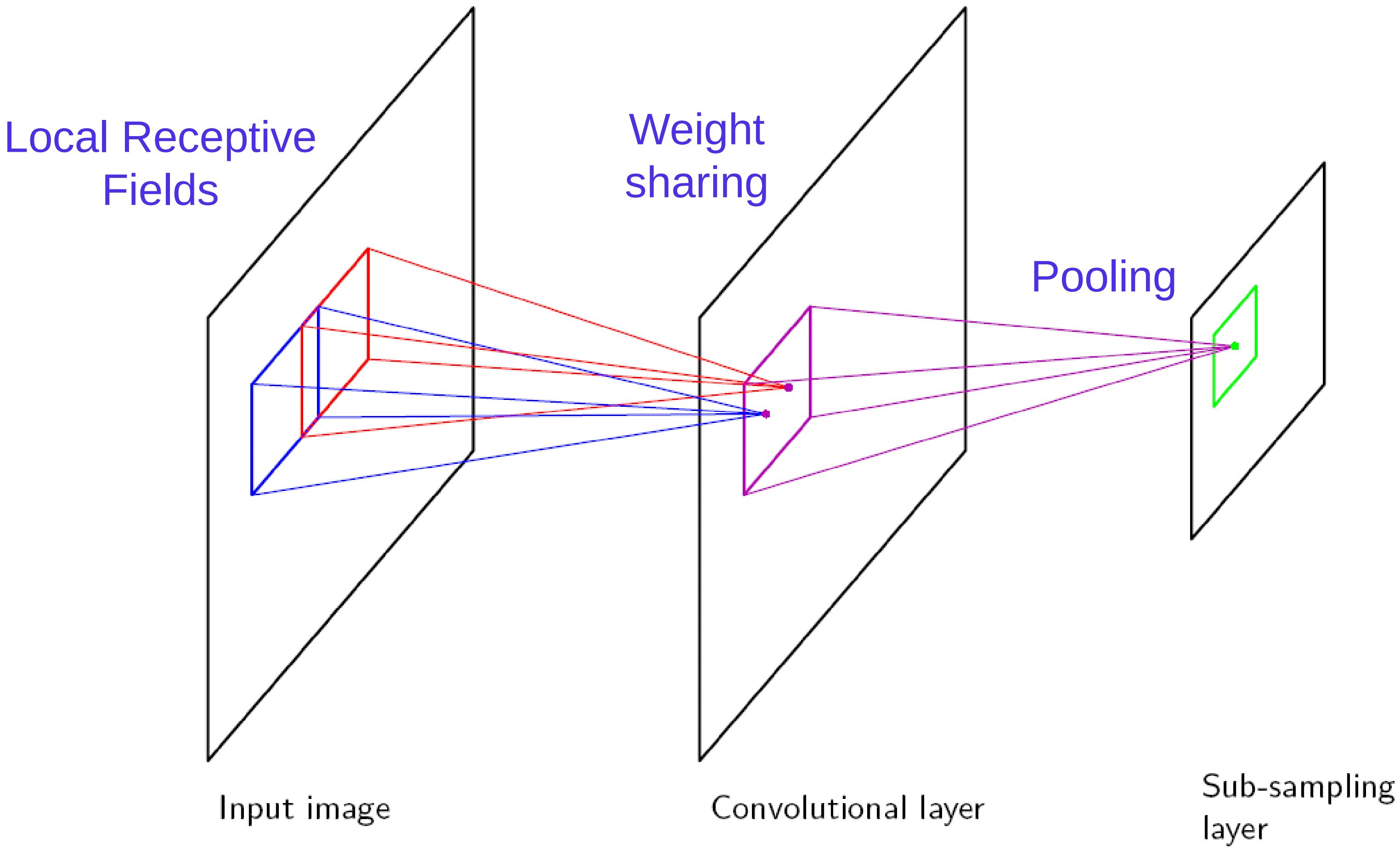
Slide Credit: Yoshua Bengio

Outline

- Restricted Boltzmann machines
- Deep Belief Networks
- Denoising Autoencoders
- Applications to Vision
- Applications to Audio and Multimodal Data

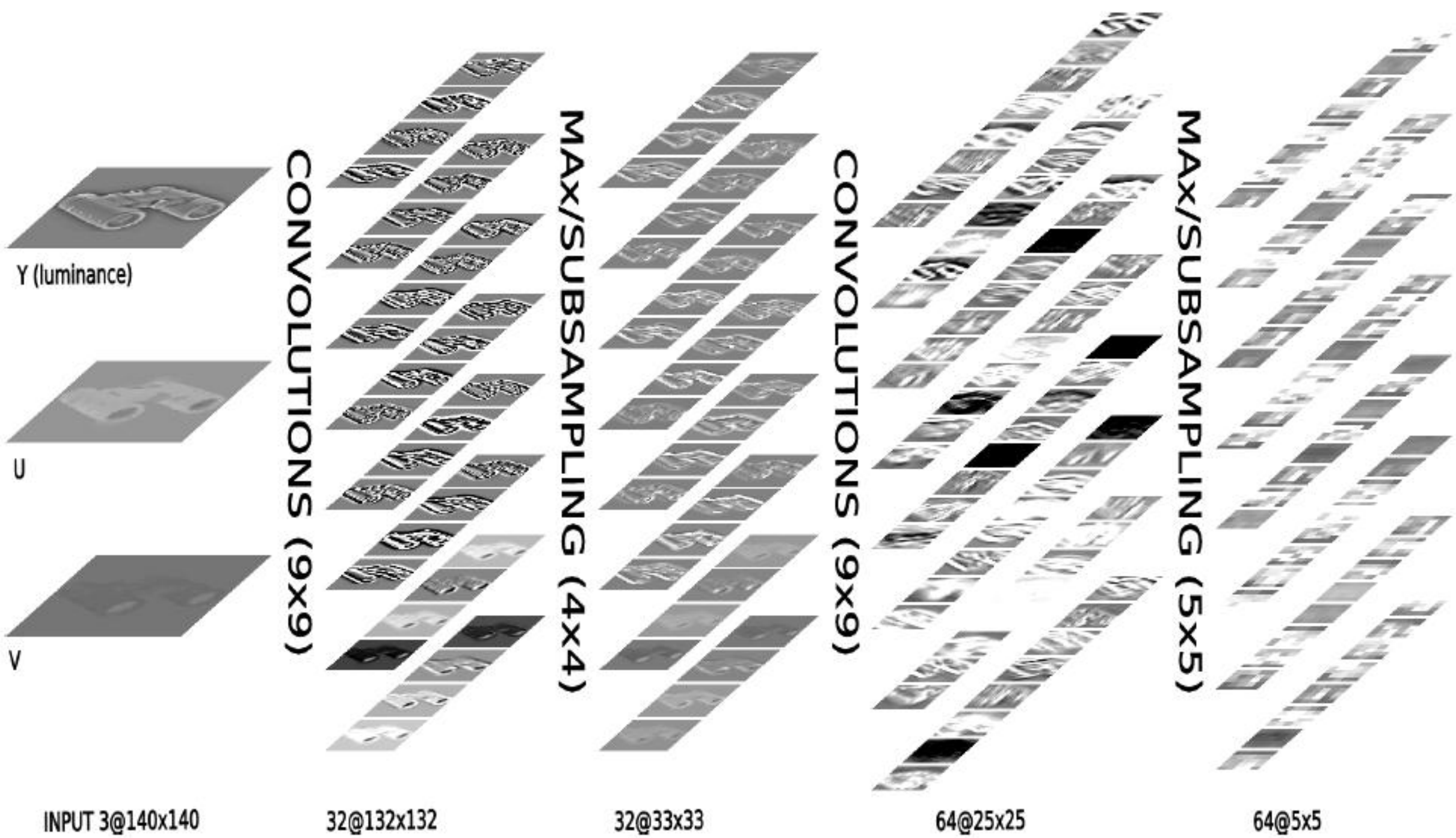
Convolutional Neural Networks

(LeCun et al., 1989)



Deep Convolutional Architectures

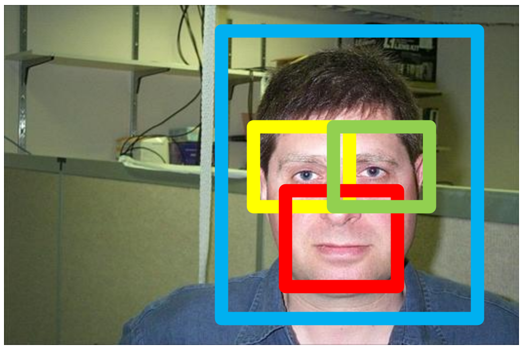
State-of-the-art on MNIST digits, Caltech-101 objects, etc.

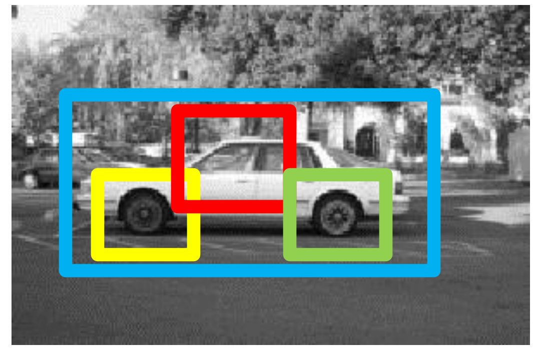


Slide Credit: Yann LeCun

Learning object representations

- Learning objects and parts in images

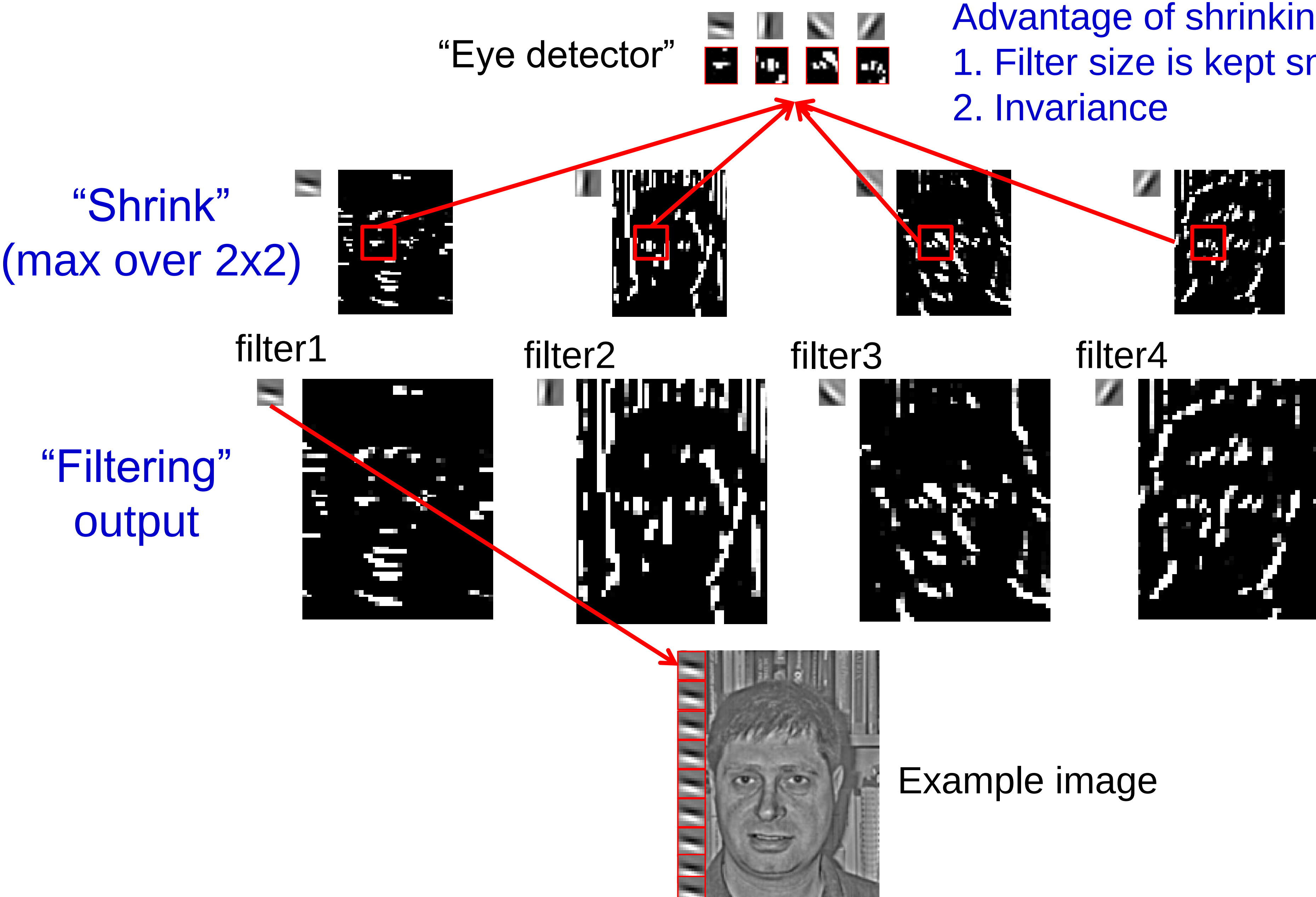




- Large image patches contain interesting higher-level structures.
  - E.g., object parts and full objects

Illustration: Learning an “eye” detector

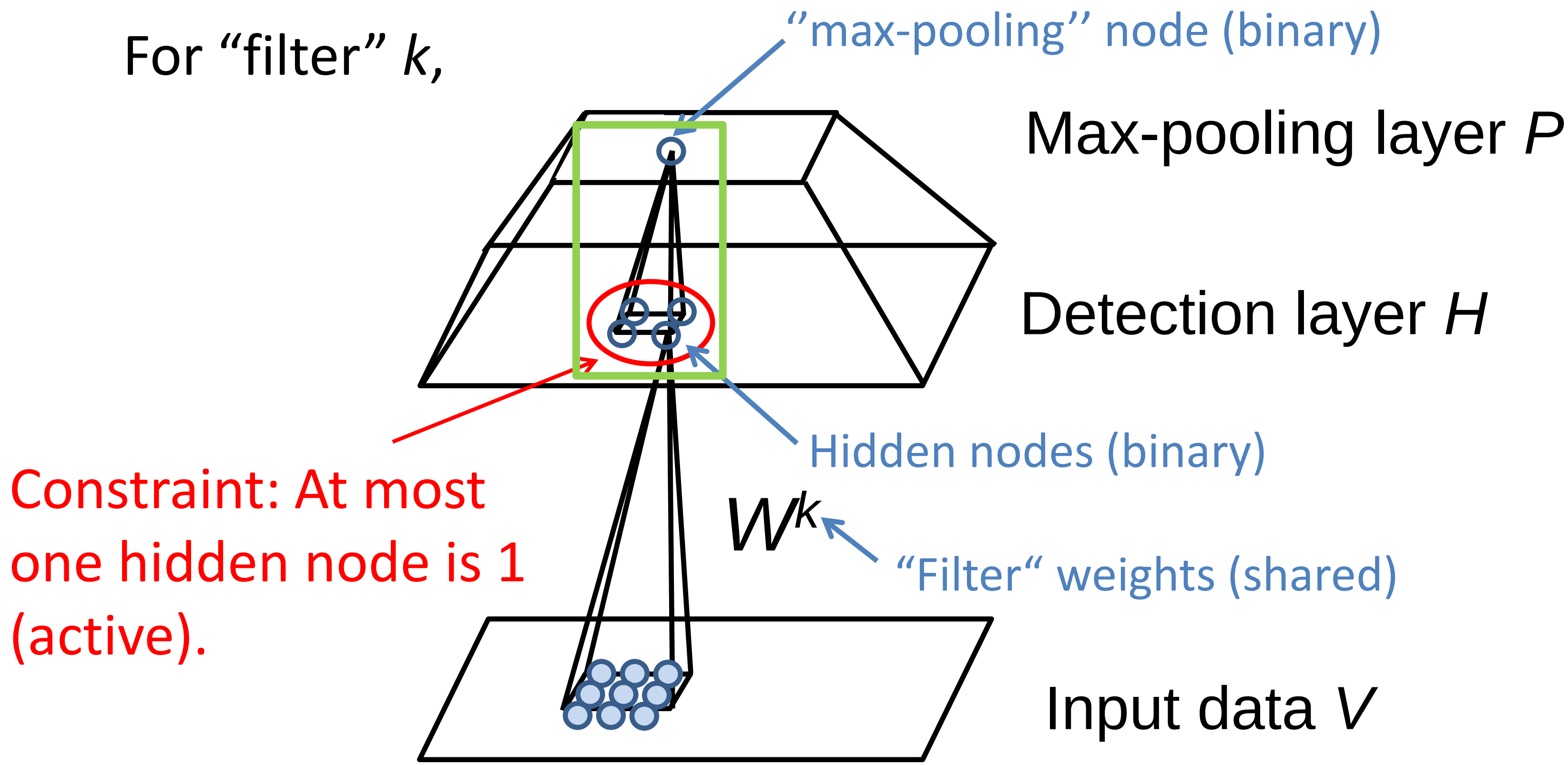
Advantage of shrinking
1. Filter size is kept sma
2. Invariance



Convolutional RBM (CRBM) [Lee et al, ICML 2009]

[Related work: Norouzi et al., CVPR 2009; Desjardins and Bengio, 2008]

For "filter" *k*,



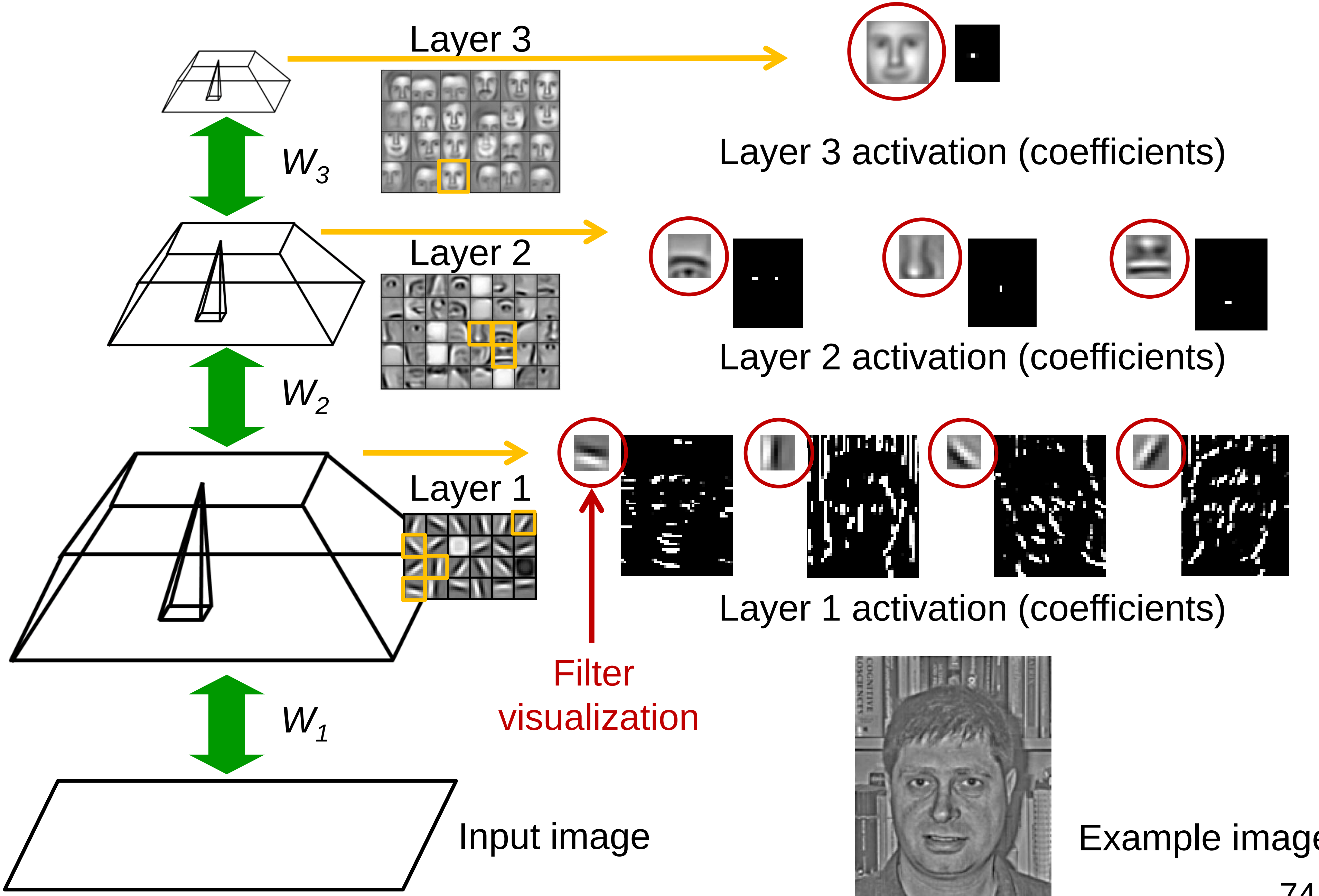
$$P(\mathbf{v},\mathbf{h}) \propto \exp\left(\sum_{i,j,k} h^k_{i,j}(\tilde{W}^k * v)_{i,j}\right)$$

$$\text{subj. to} \quad \sum_{(i,j)\in\text{“}cell(y)\text{”}} h^k_{i,j} \le 1, \forall k, y.$$

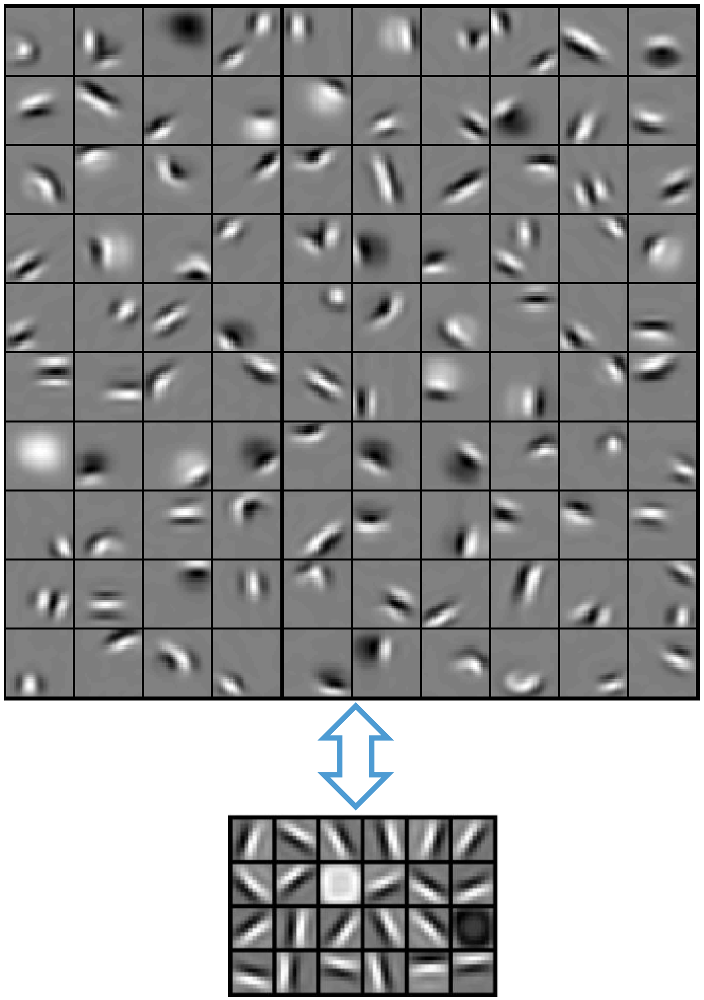
Key Properties

- Convolutional structure
- Probabilistic max-pooling ("mutual exclusion")

Convolutional deep belief networks illustration



Unsupervised learning from natural images



Second layer bases

contours, corners, arcs, surface boundaries

First layer bases

localized, oriented edges

Unsupervised learning of object-parts

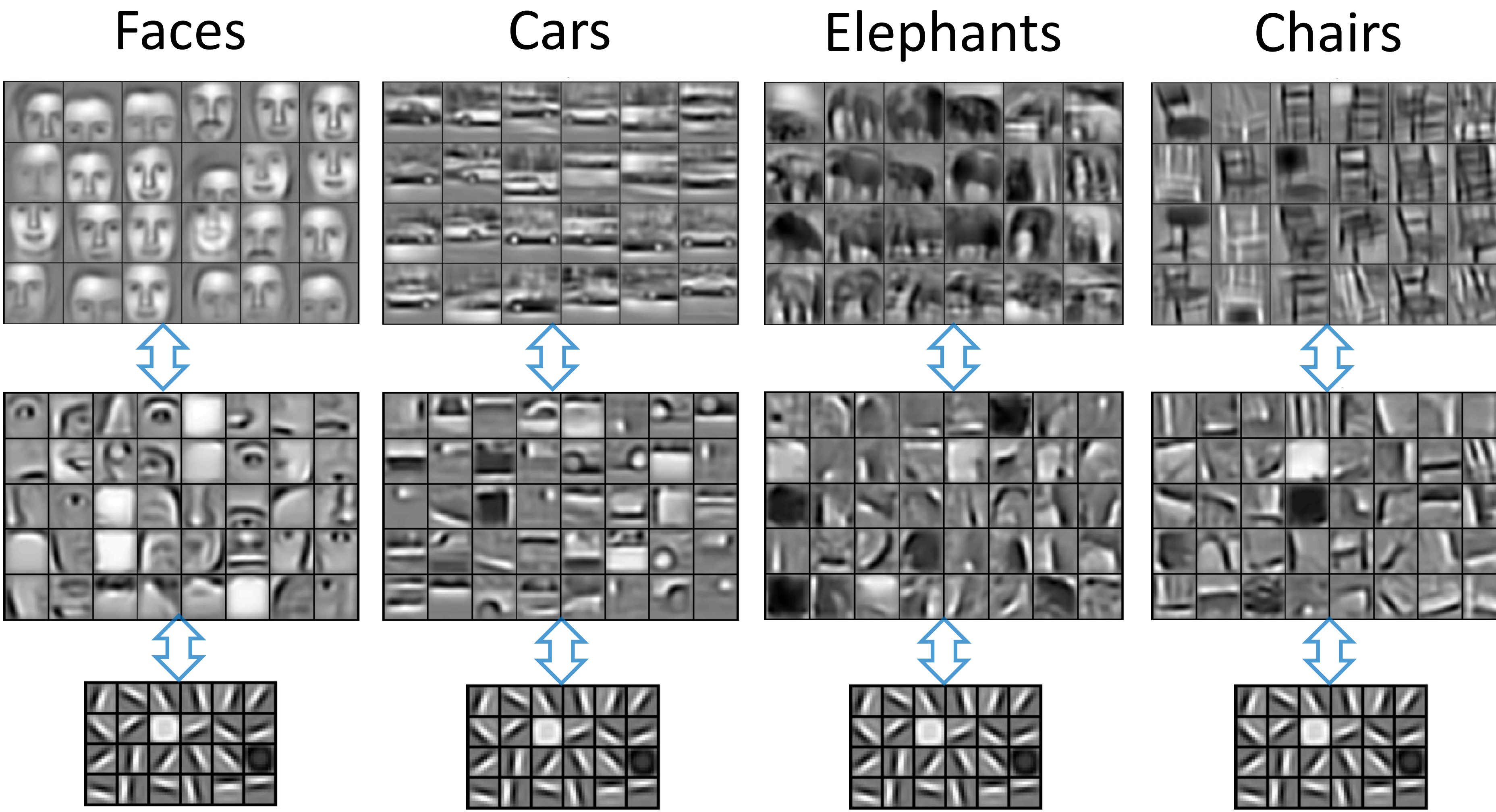
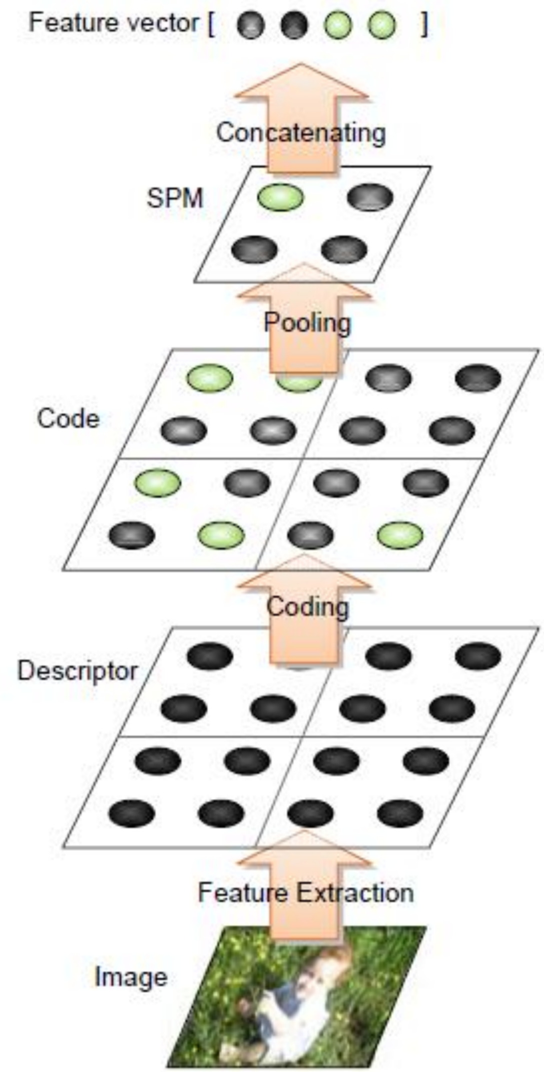


Image classification with Spatial Pyramids

[Lazebnik et al., CVPR 2005; Yang et al., CVPR 2009]

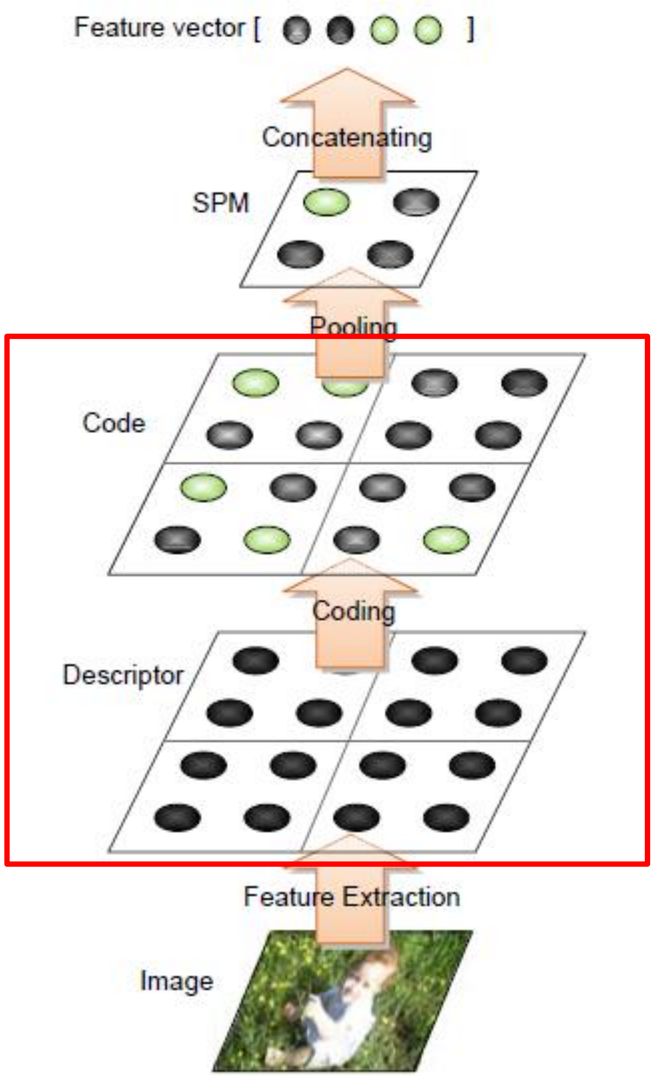
- **Descriptor Layer:** detect and locate features, extract corresponding descriptors (e.g. SIFT)
- **Code Layer:** code the descriptors
  - Vector Quantization (VQ): each code has only one non-zero element
  - Soft-VQ: small group of elements can be non-zero
- **SPM layer:** pool codes across subregions and average/normalize into a histogram



Slide Credit: Kai Yu

Improving the coding step

- Classifiers using these features need nonlinear kernels
  - Lazebnik et al., CVPR 2005; Grauman and Darrell, JMLR 2007
  - High computational complexity
- Idea: modify the <u>coding step</u> to produce feature representations that linear classifiers can use effectively
  - **Sparse coding** [Olshausen & Field, Nature 1996; Lee et al., NIPS 2007; Yang et al., CVPR 2009; Boureau et al., CVPR 2010]
  - **Local Coordinate coding** [Yu et al., NIPS 2009; Wang et al., CVPR 2010]
  - **RBMs** [Sohn, Jung, Lee, Hero III, ICCV 2011]
  - Other feature learning algorithms



Slide Credit: Kai Yu

Object Recognition Results

- Classification accuracy on Caltech 101/256

< Caltech 101 >

# of training images	15	30
Zhang et al., CVPR 2005	59.1	66.2
Griffin et al., 2008	59.0	67.6
ScSPM [Yang et al., CVPR 2009]	67.0	73.2
LLC [Wang et al., CVPR 2010]	65.4	73.4
Macrofeatures [Boureau et al., CVPR 2010]	-	75.7
Boureau et al., ICCV 2011	-	77.1
Sparse RBM [Sohn et al., ICCV 2011]	68.6	74.9
Sparse CRBM [Sohn et al., ICCV 2011]	71.3	77.8

Competitive performance to other state-of-the-art methods using a single type of features

Object Recognition Results

- Classification accuracy on Caltech 101/256

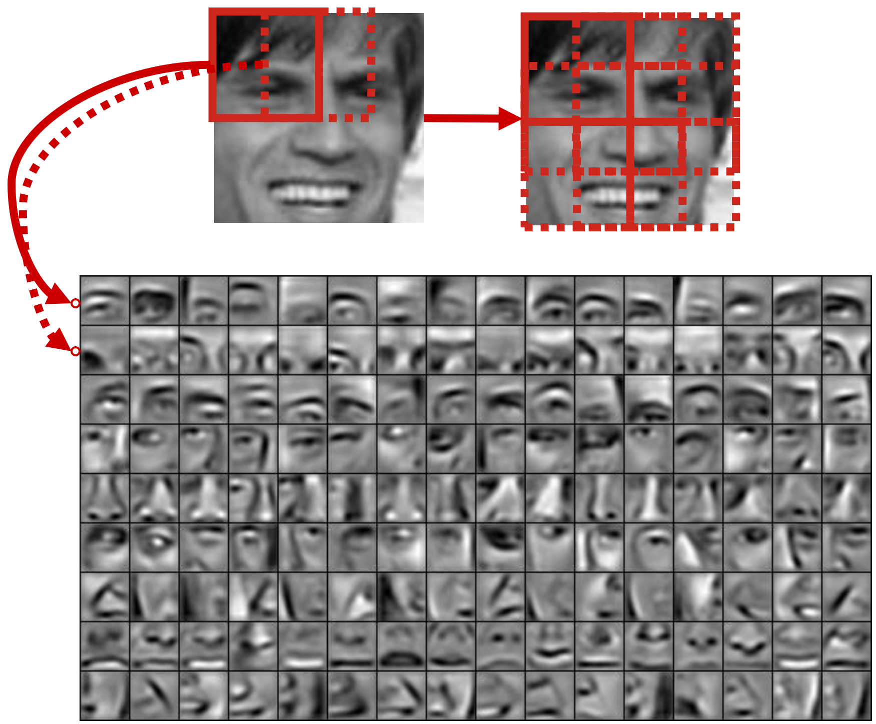
< Caltech 256 >

# of training images	30	60
Griffin et al. [2]	34.10	-
vanGemert et al., PAMI 2010	27.17	-
ScSPM [Yang et al., CVPR 2009]	34.02	40.14
LLC [Wang et al., CVPR 2010]	41.19	47.68
Sparse CRBM [Sohn et al., ICCV 2011]	42.05	47.94

Competitive performance to other state-of-the-art methods
using a single type of features

Local Convolutional RBM

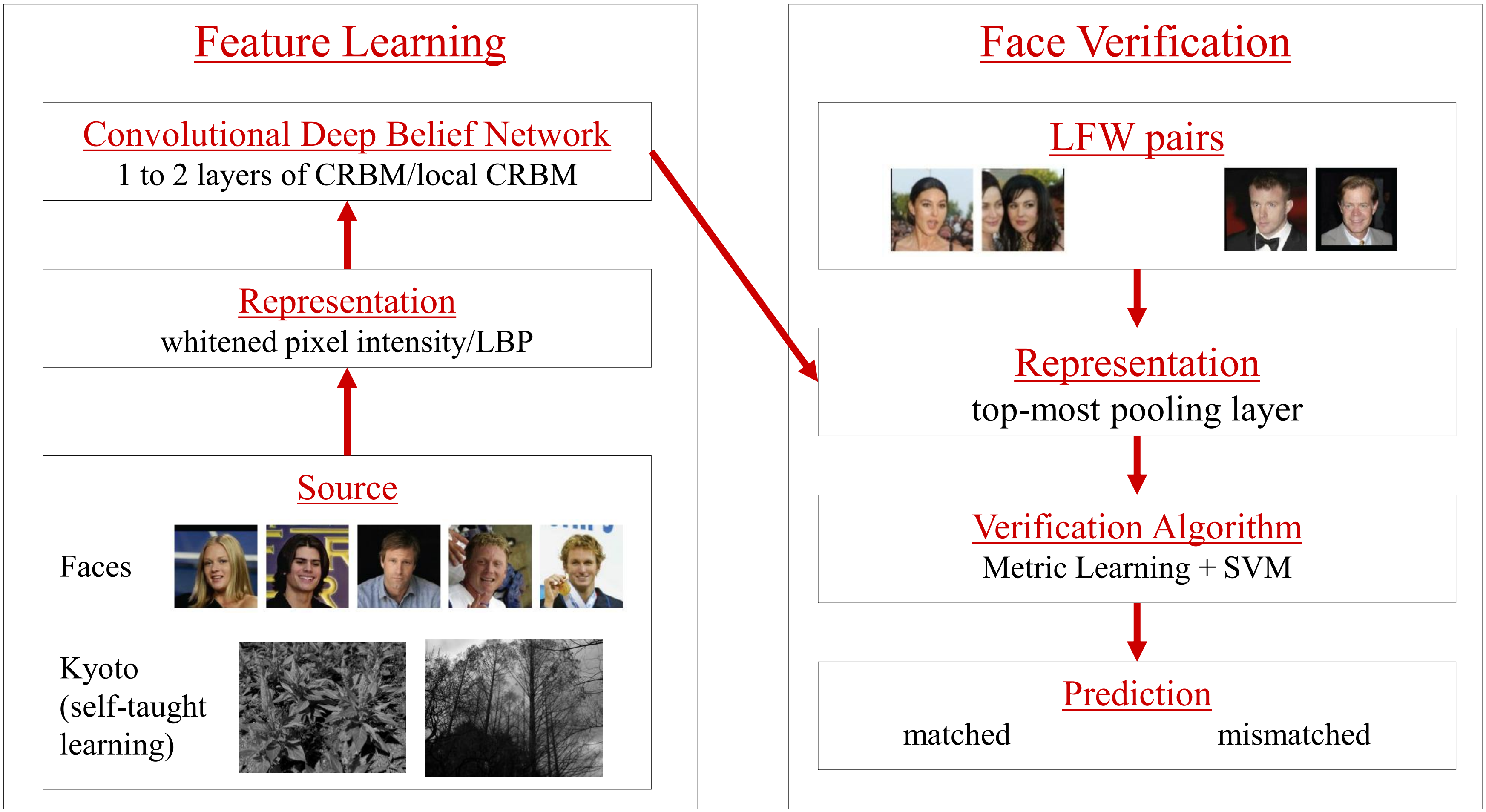
- Modeling convolutional structures in local regions jointly
  - More statistically effective in learning for non-stationary (roughly aligned) images



[Huang, Lee, Learned-Miller, CVPR 2012]

Face Verification

[Huang, Lee, Learned-Miller, CVPR 2012]



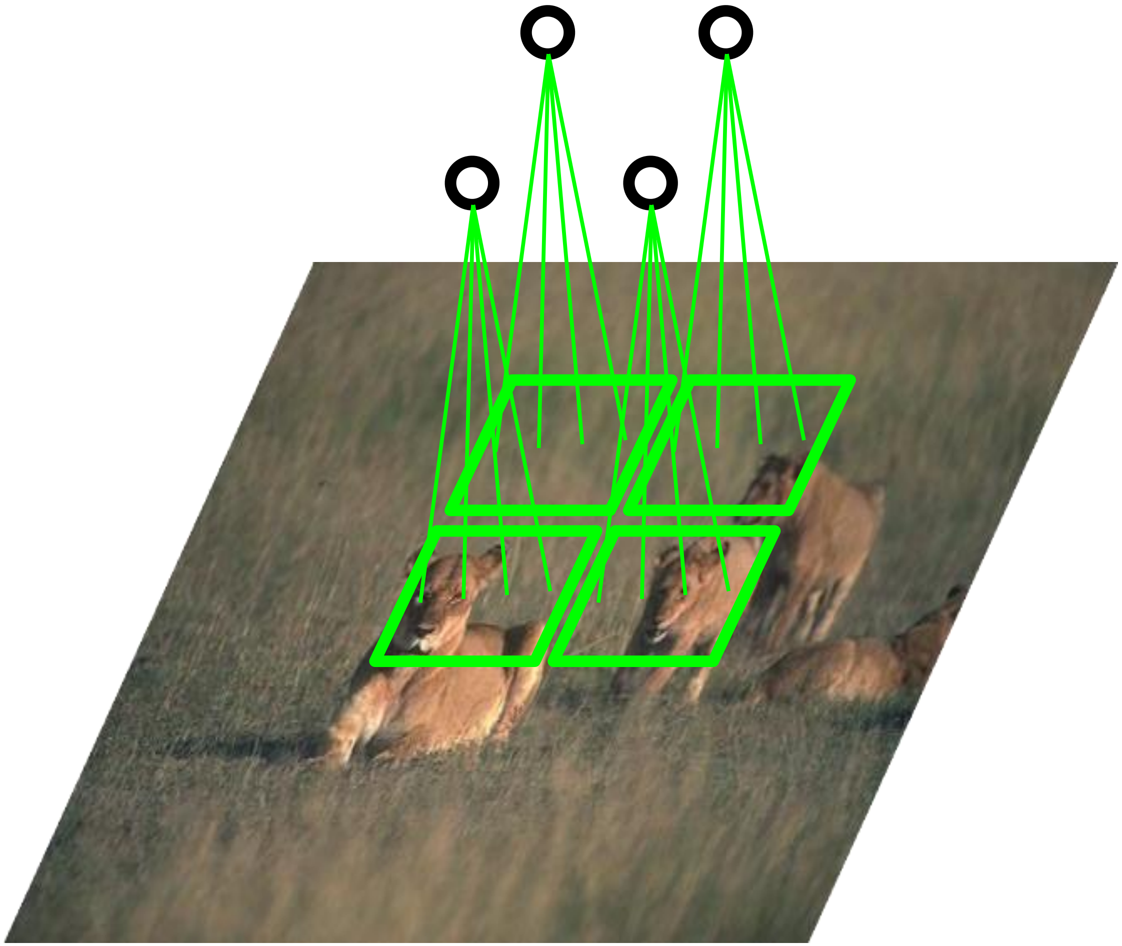
Face Verification

[Huang, Lee, Learned-Miller, CVPR 2012]

Method	Accuracy ± SE
V1-like with MKL (Pinto et al., CVPR 2009)	0.7935 ± 0.0055
Linear rectified units (Nair and Hinton, ICML 2010)	0.8073 ± 0.0134
CSML (Nguyen & Bai, ACCV 2010)	0.8418 ± 0.0048
Learning-based descriptor (Cao et al., CVPR 2010)	0.8445 ± 0.0046
OSS, TSS, full (Wolf et al., ACCV 2009)	0.8683 ± 0.0034
OSS only (Wolf et al., ACCV 2009)	0.8207 ± 0.0041
Combined (LBP + deep learning features)	0.8777 ± 0.0062

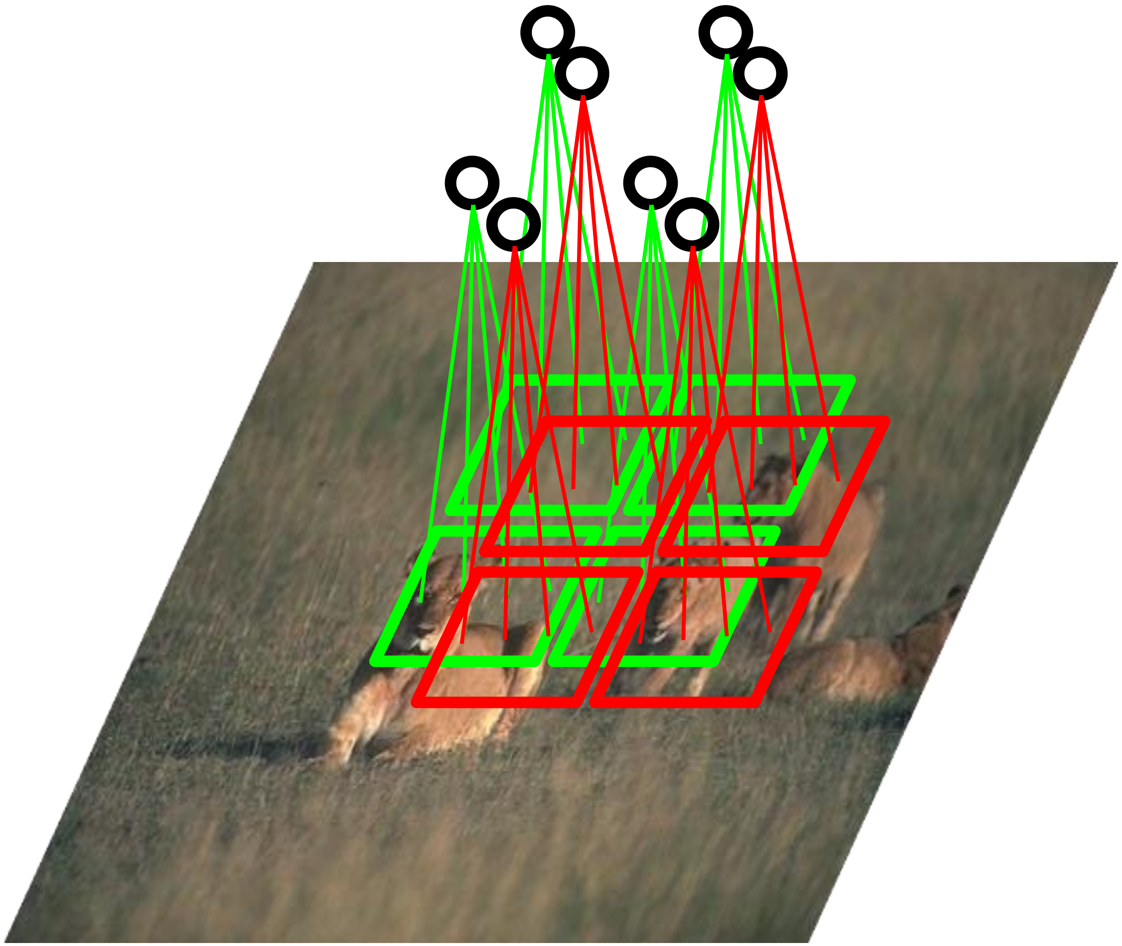
Tiled Convolutional models

IDEA: have one subset of filters applied to these locations,



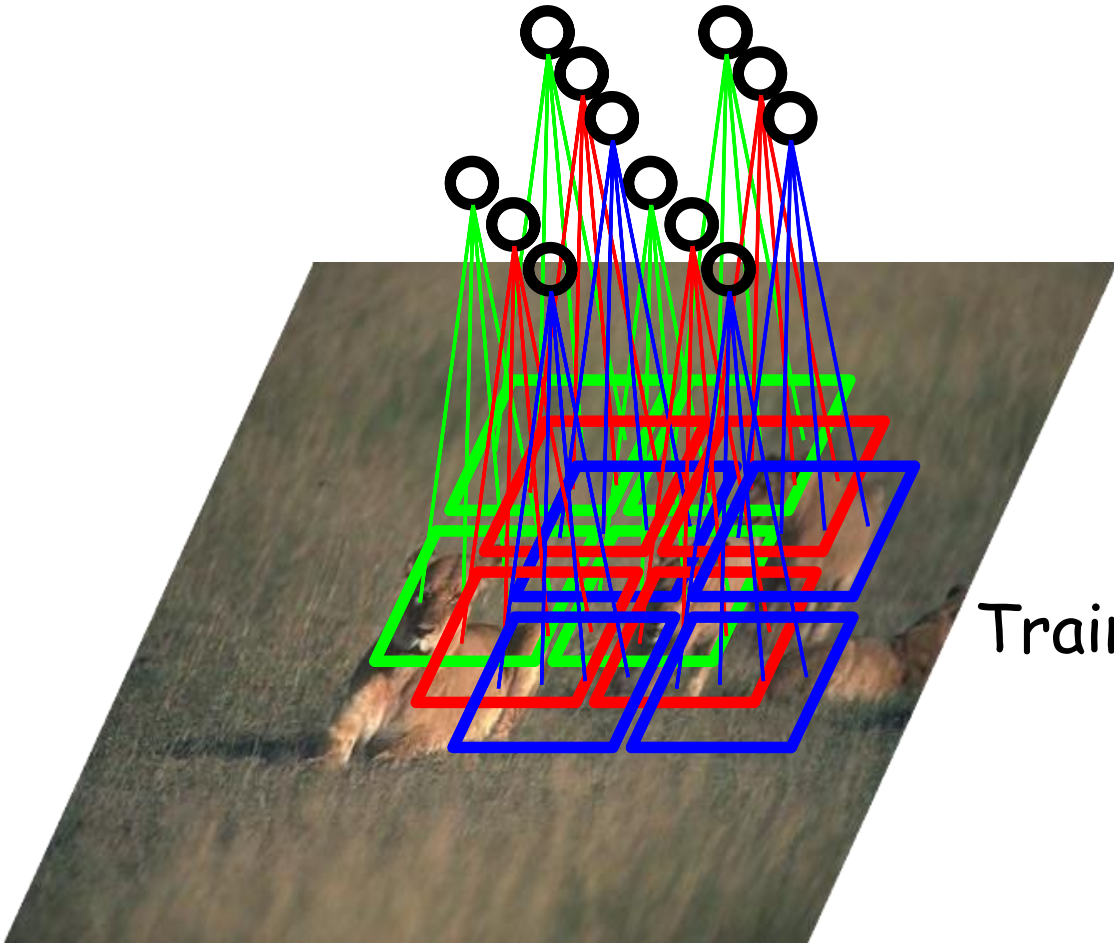
Tiled Convolutional models

IDEA: have one subset of filters applied to these locations, another subset to these locations



Tiled Convolutional models

IDEA: have one subset of filters applied to these locations, another subset to these locations, etc.



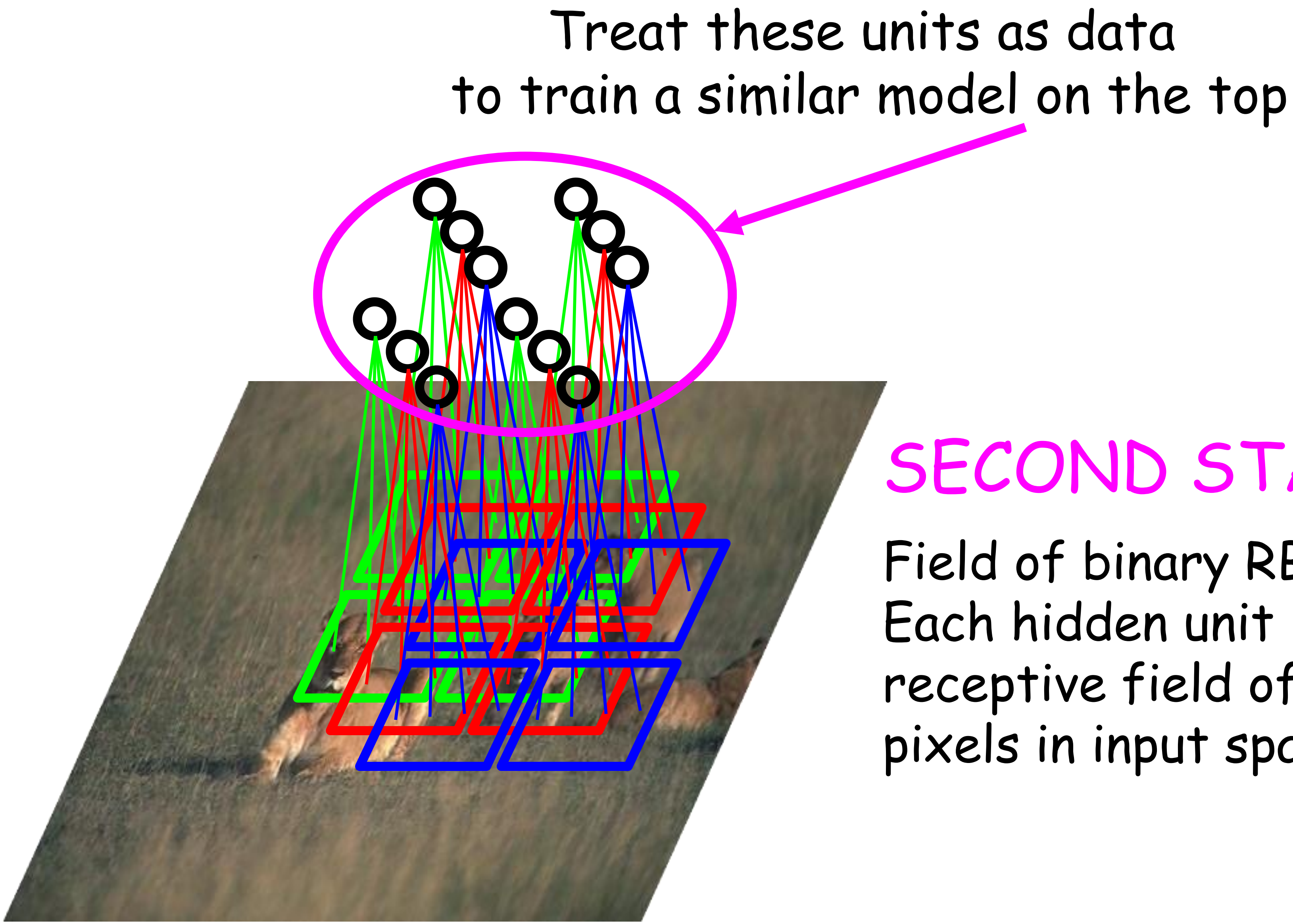
Train jointly all parameters.

No block artifacts
Reduced redundancy

Gregor LeCun arXiv 2010
Ranzato, Mnih, Hinton NIPS 2010

93

Tiled Convolutional models



SECOND STAGE

Field of binary RBM's.
Each hidden unit has a receptive field of 30x30 pixels in input space.

94

Facial Expression Recognition

Toronto Face Dataset (J. Susskind et al. 2010)

~ 100K unlabeled faces from different sources

~ 4K labeled images

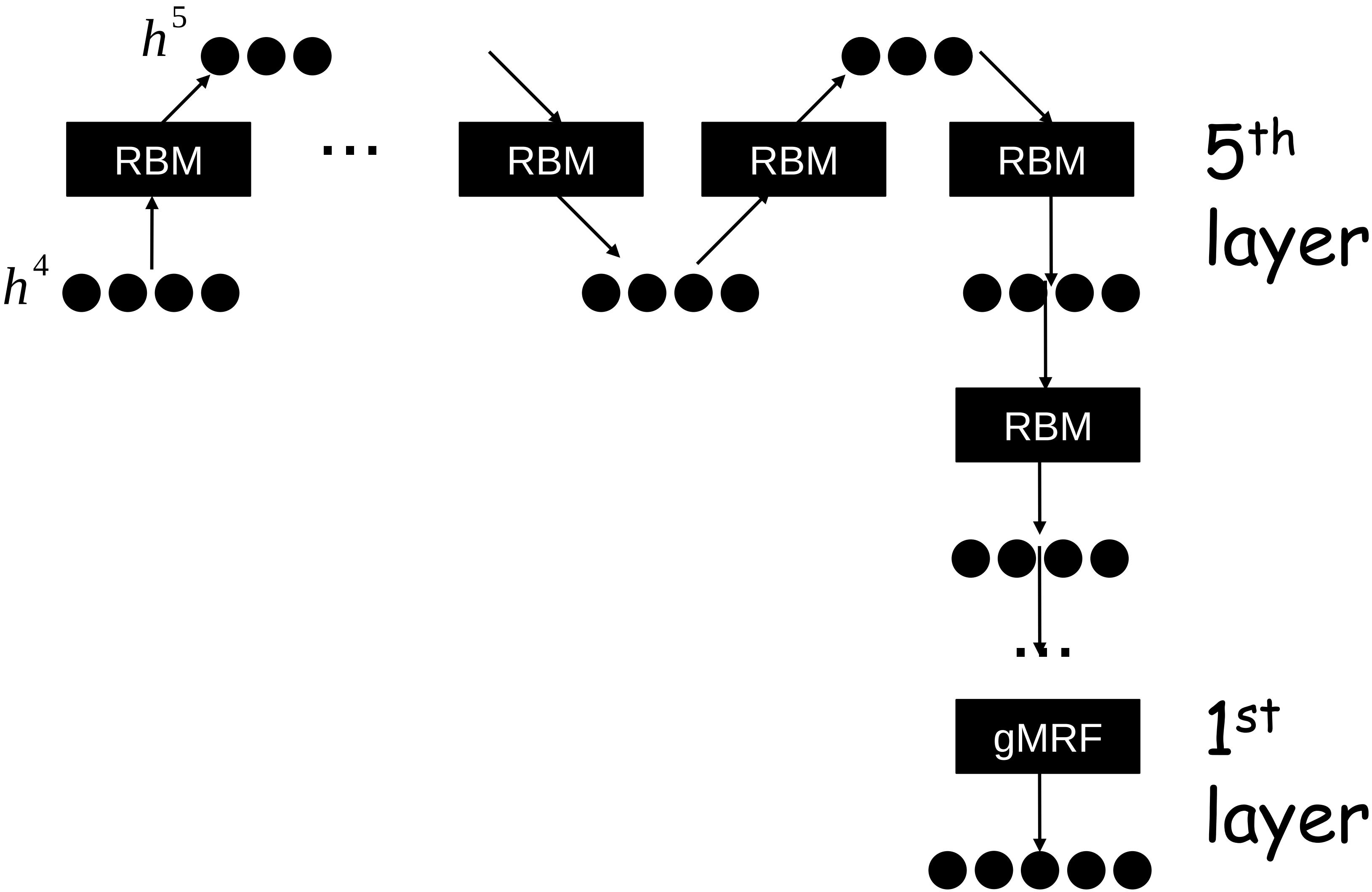
Resolution: 48x48 pixels

7 facial expressions

Ranzato, et al. CVPR 2011

Facial Expression Recognition

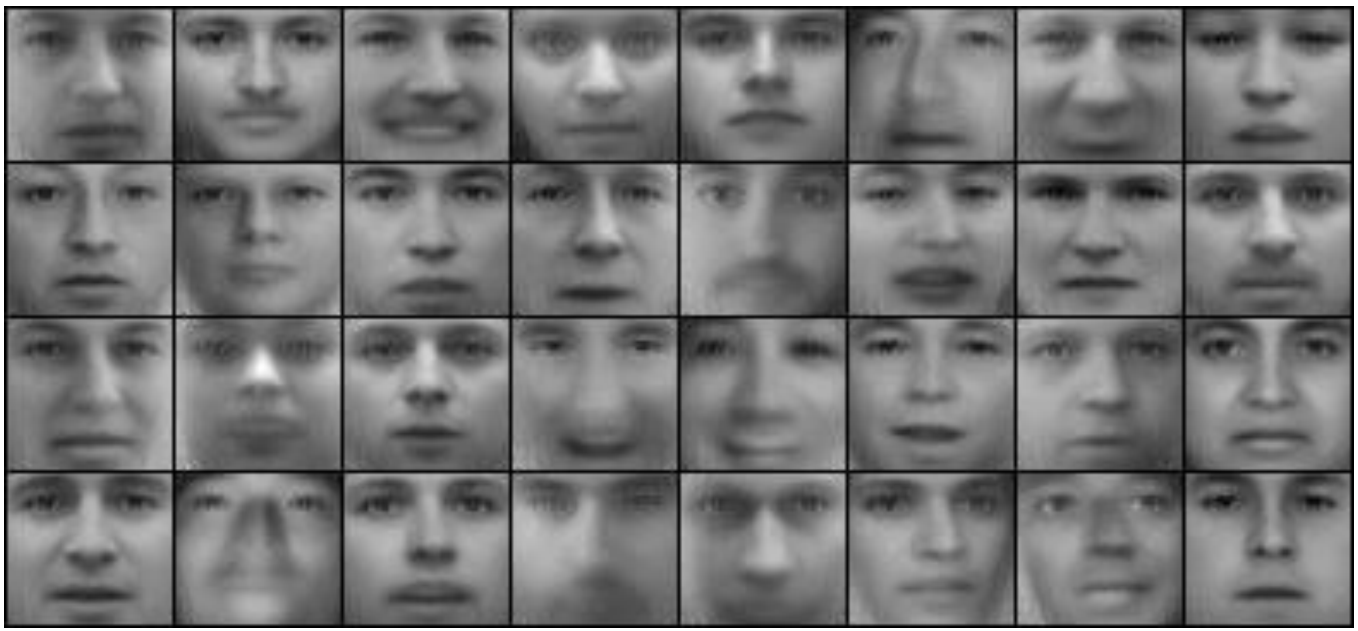
Drawing samples from the model (5^{th} layer with 128 hiddens)



96

Facial Expression Recognition

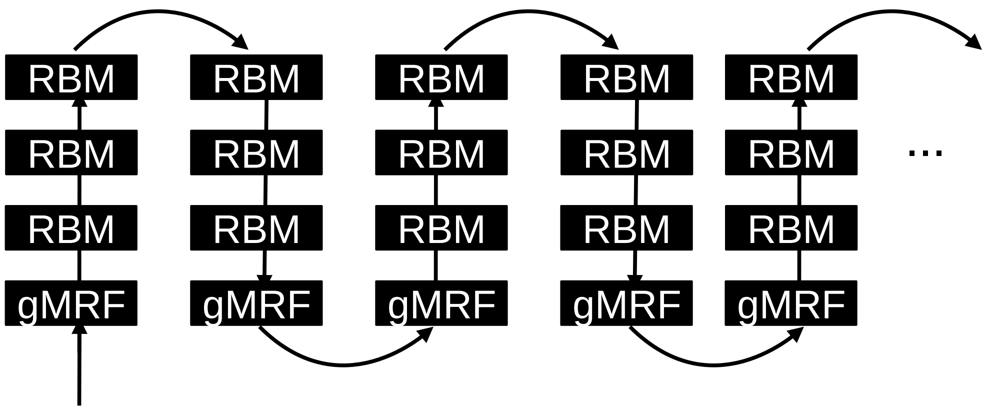
Drawing samples from the model (5^{th} layer with 128 hiddens)



Ranzato, et al. CVPR 2011

Facial Expression Recognition

- 7 synthetic occlusions
- use generative model to fill-in
  (conditional on the known pixels)



Ranzato, et al. CVPR 2011

Facial Expression Recognition

originals

Type 1 occlusion: eyes

Restored images

Ranzato, et al. CVPR 2011

Facial Expression Recognition

originals

Type 2 occlusion: mouth

Restored images

Ranzato, et al. CVPR 2011

Facial Expression Recognition

originals

Type 3 occlusion: right half

Restored images

Ranzato, et al. CVPR 2011

Facial Expression Recognition

originals

Type 5 occlusion: top half

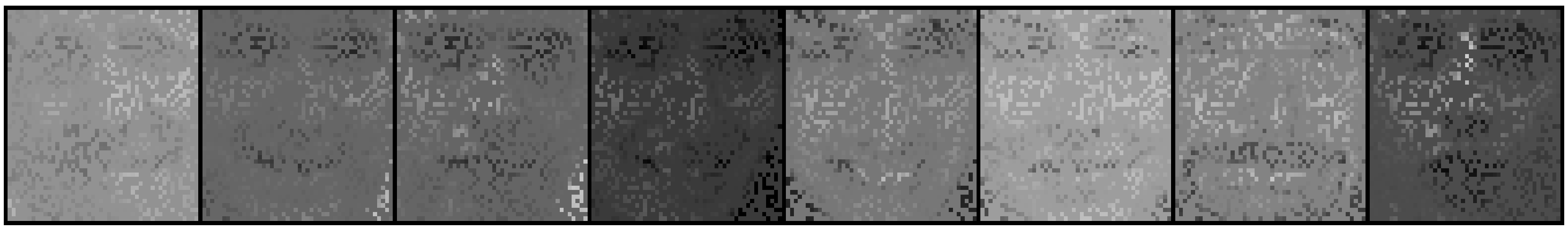
Restored images

Ranzato, et al. CVPR 2011

Facial Expression Recognition

originals

Type 7 occlusion: 70% of pixels at random

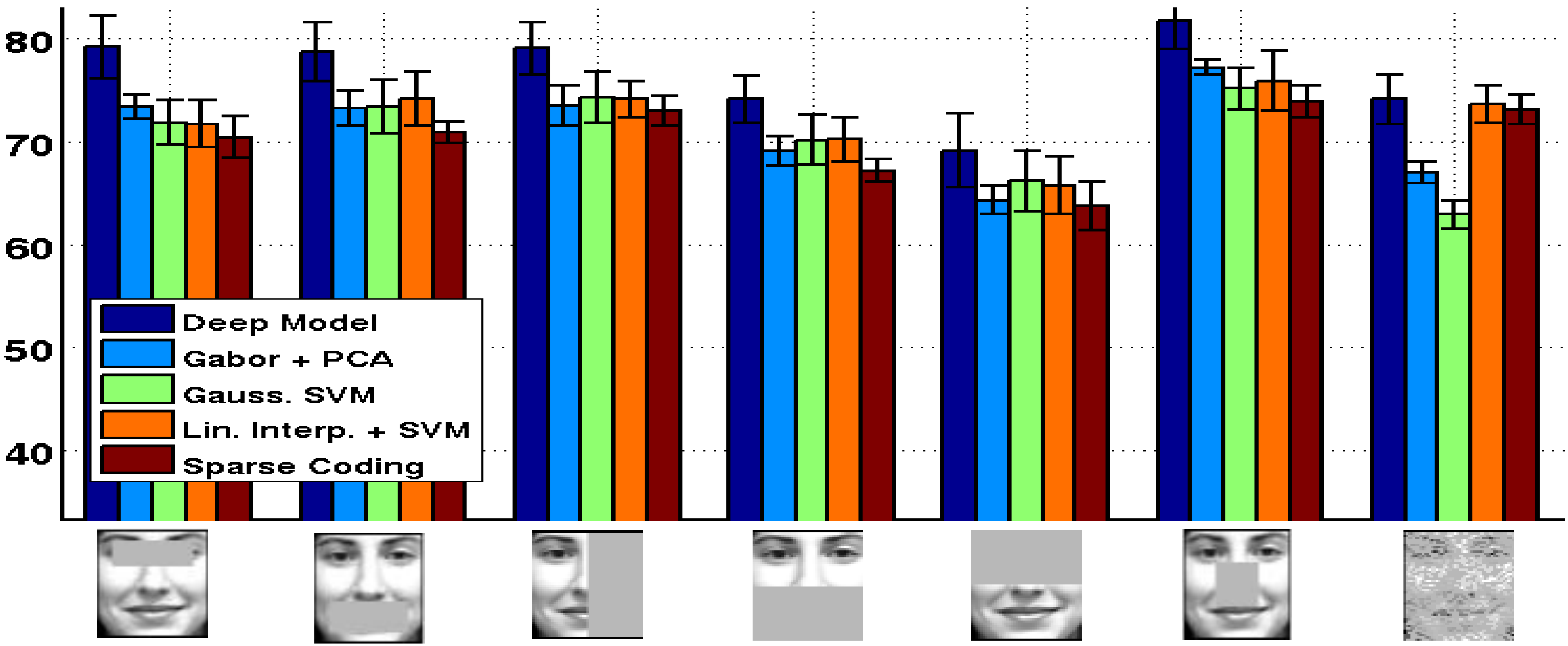


Restored images

Ranzato, et al. CVPR 2011

Facial Expression Recognition

occluded images for both training and test



Dailey, et al. J. Cog. Neuros. 2003

Wright, et al. PAMI 2008

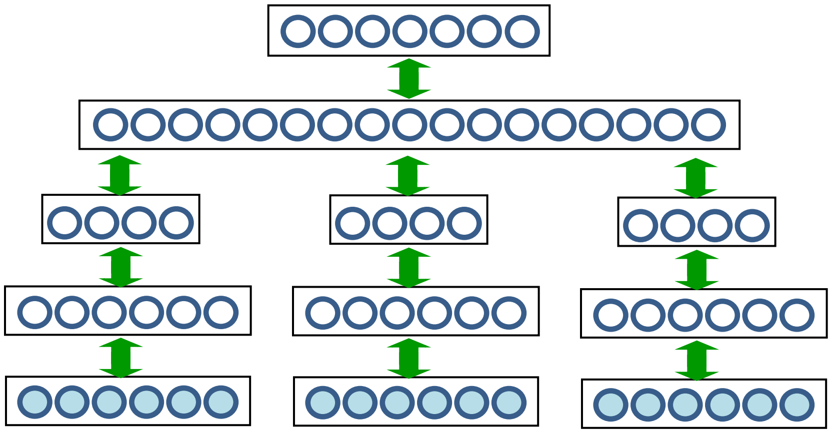
Ranzato, et al. CVPR 2011

Outline

- Restricted Boltzmann machines
- Deep Belief Networks
- Denoising Autoencoders
- Applications to Vision
- Applications to Audio and Multimodal Data

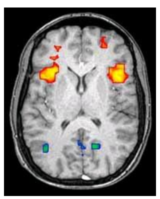
Motivation: Multi-modal learning

- Single learning algorithms that combine multiple input domains
  - Images
  - Audio & speech
  - Video
  - Text
  - Robotic sensors
  - Time-series data
  - Others

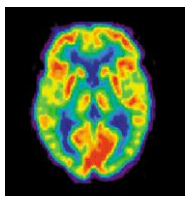


Motivation: Multi-modal learning

- Benefits: more robust performance in
  - **Multimedia processing**
  - **Biomedical data mining**

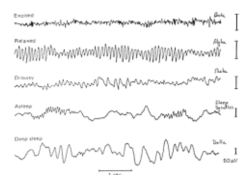


fMRI



PET scan

X-ray



EEG

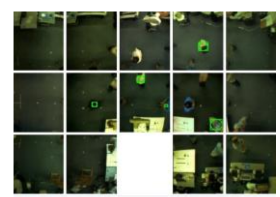
Ultra sound

  - **Robot perception**

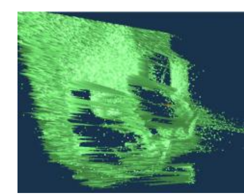
Visible light image

Audio

Thermal Infrared

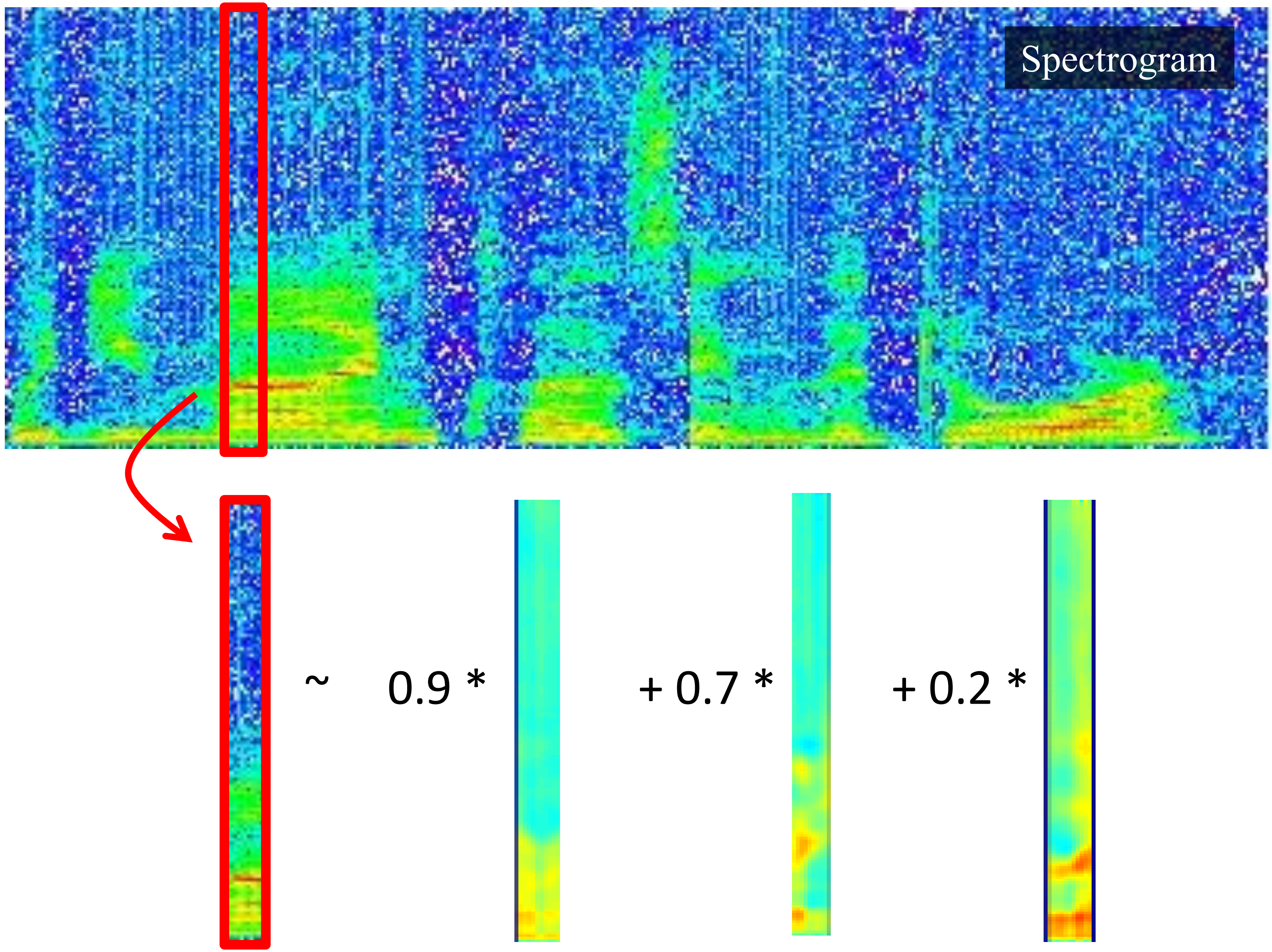


Camera array



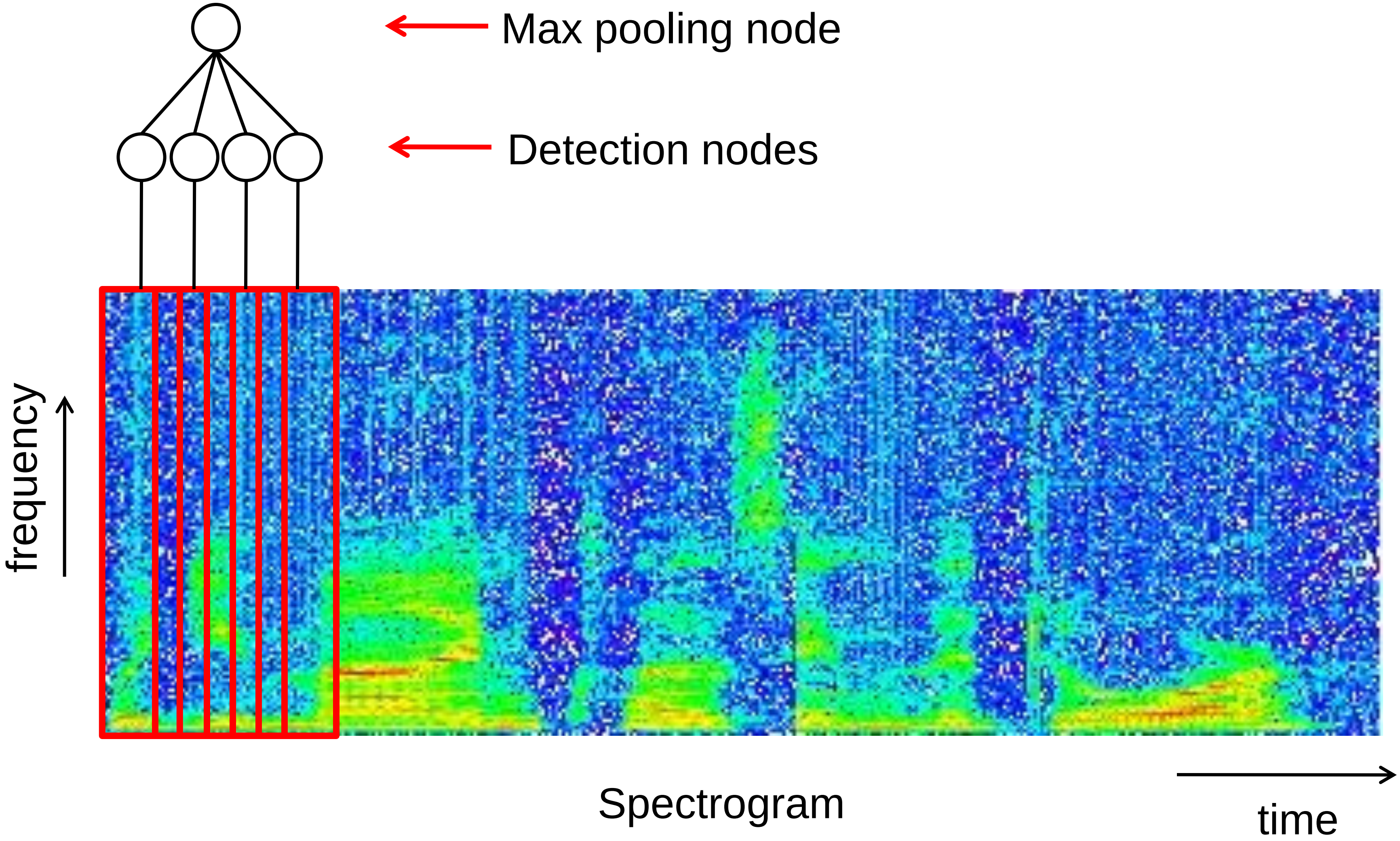
3d range scans

Sparse dictionary learning on audio



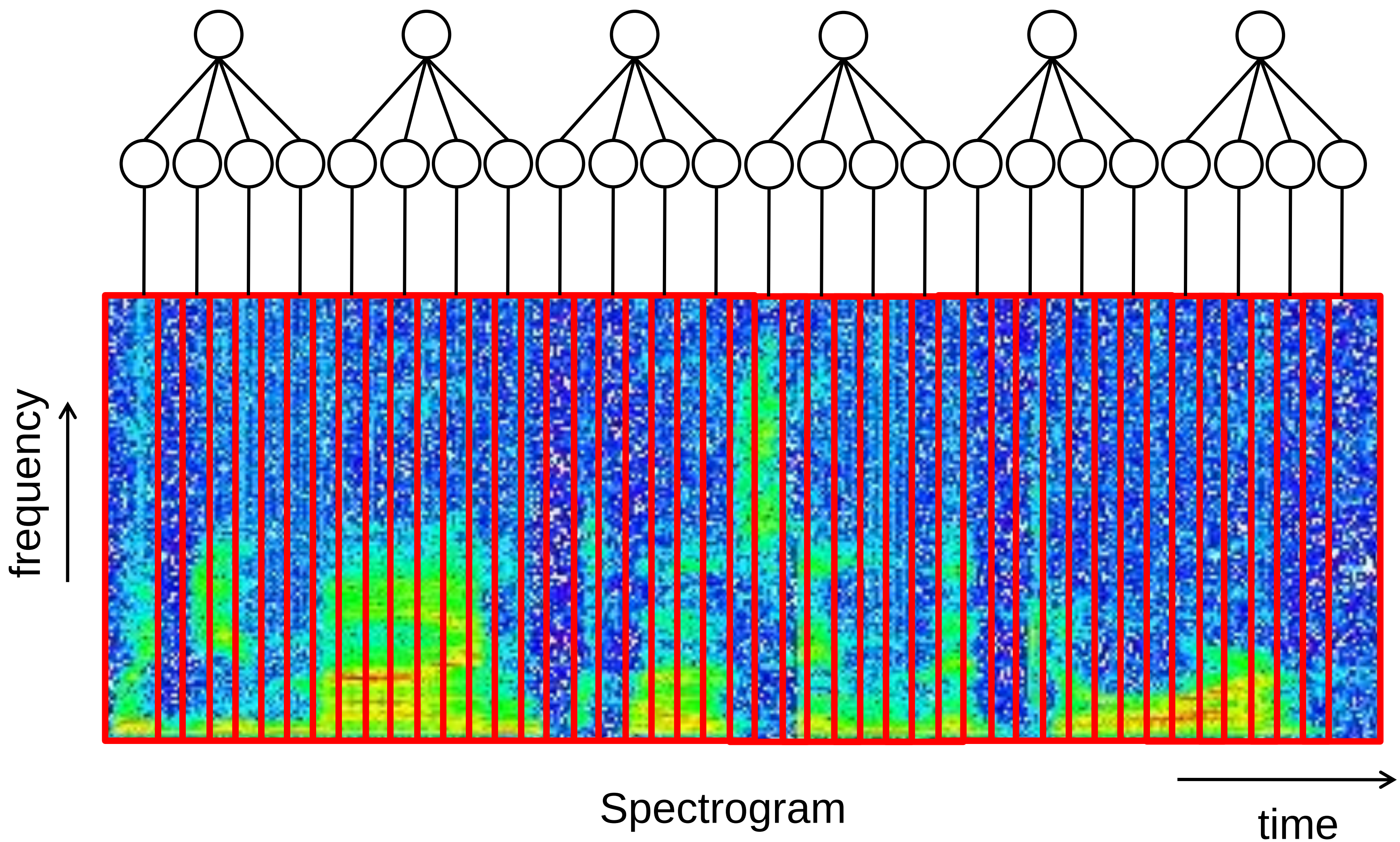
[Lee, Largman, Pham, Ng, NIPS 2009]

Convolutional DBN for audio



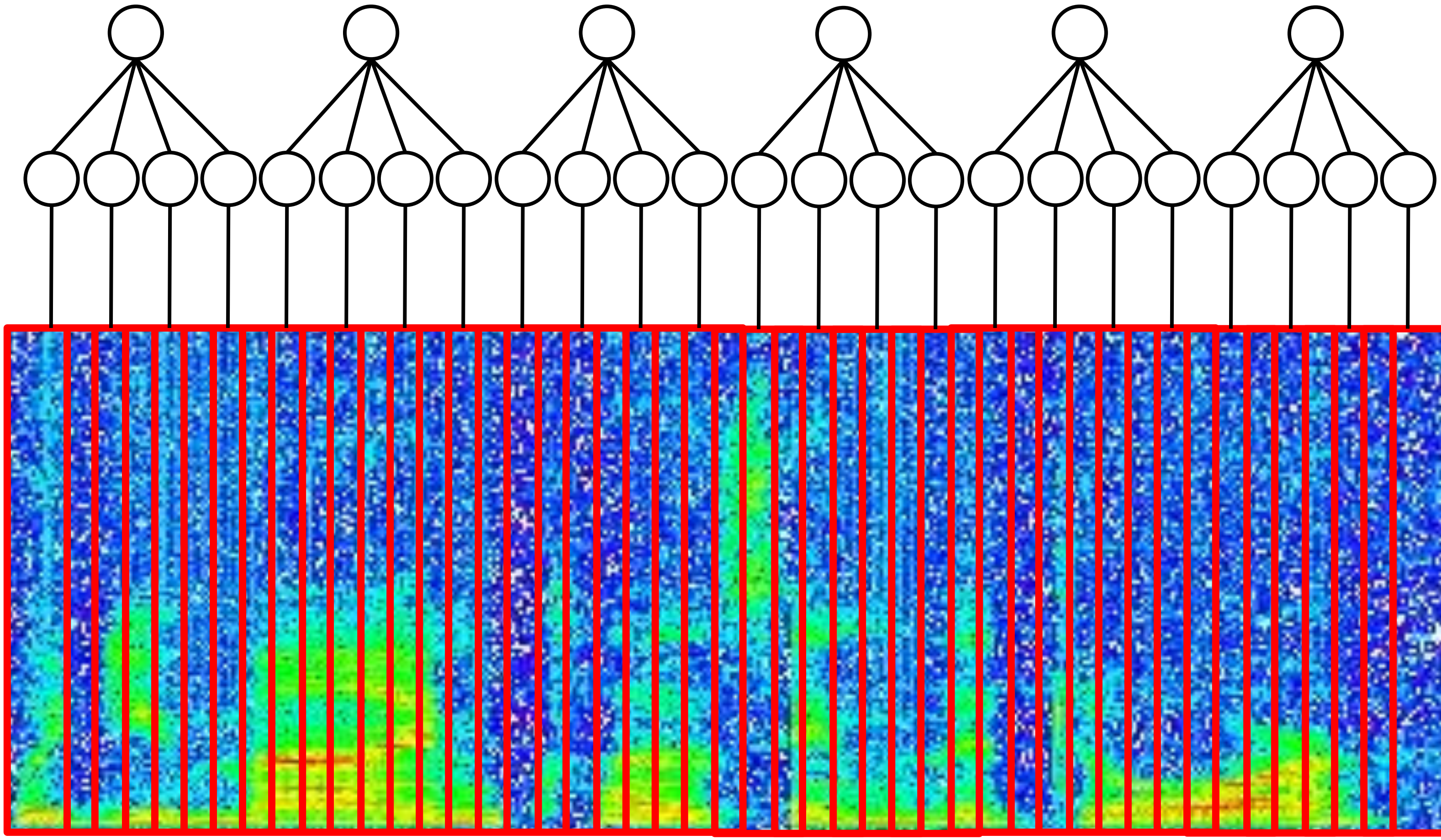
[Lee, Largman, Pham, Ng, NIPS 2009]

Convolutional DBN for audio



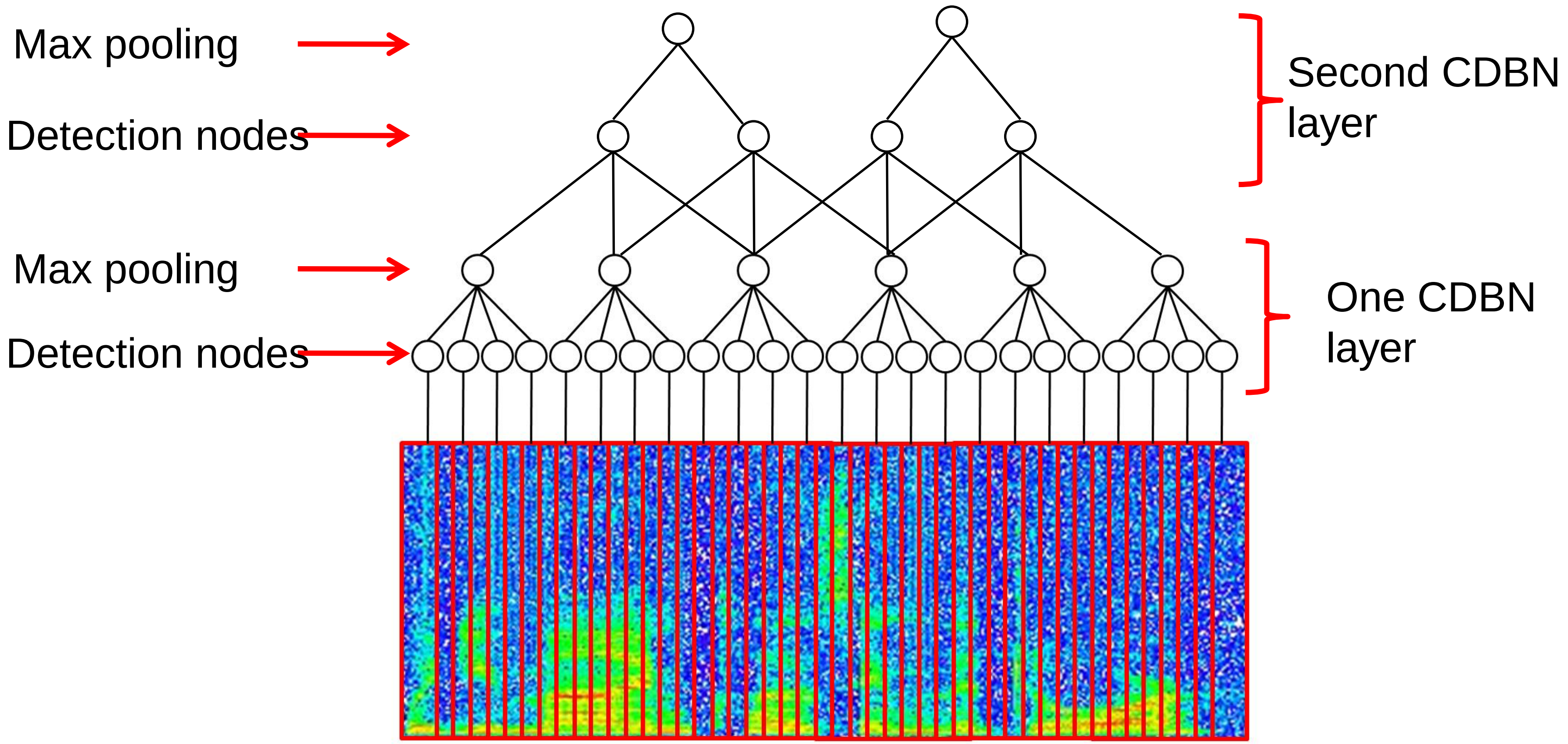
[Lee, Largman, Pham, Ng, NIPS 2009]

Convolutional DBN for audio



[Lee, Largman, Pham, Ng, NIPS 2009]

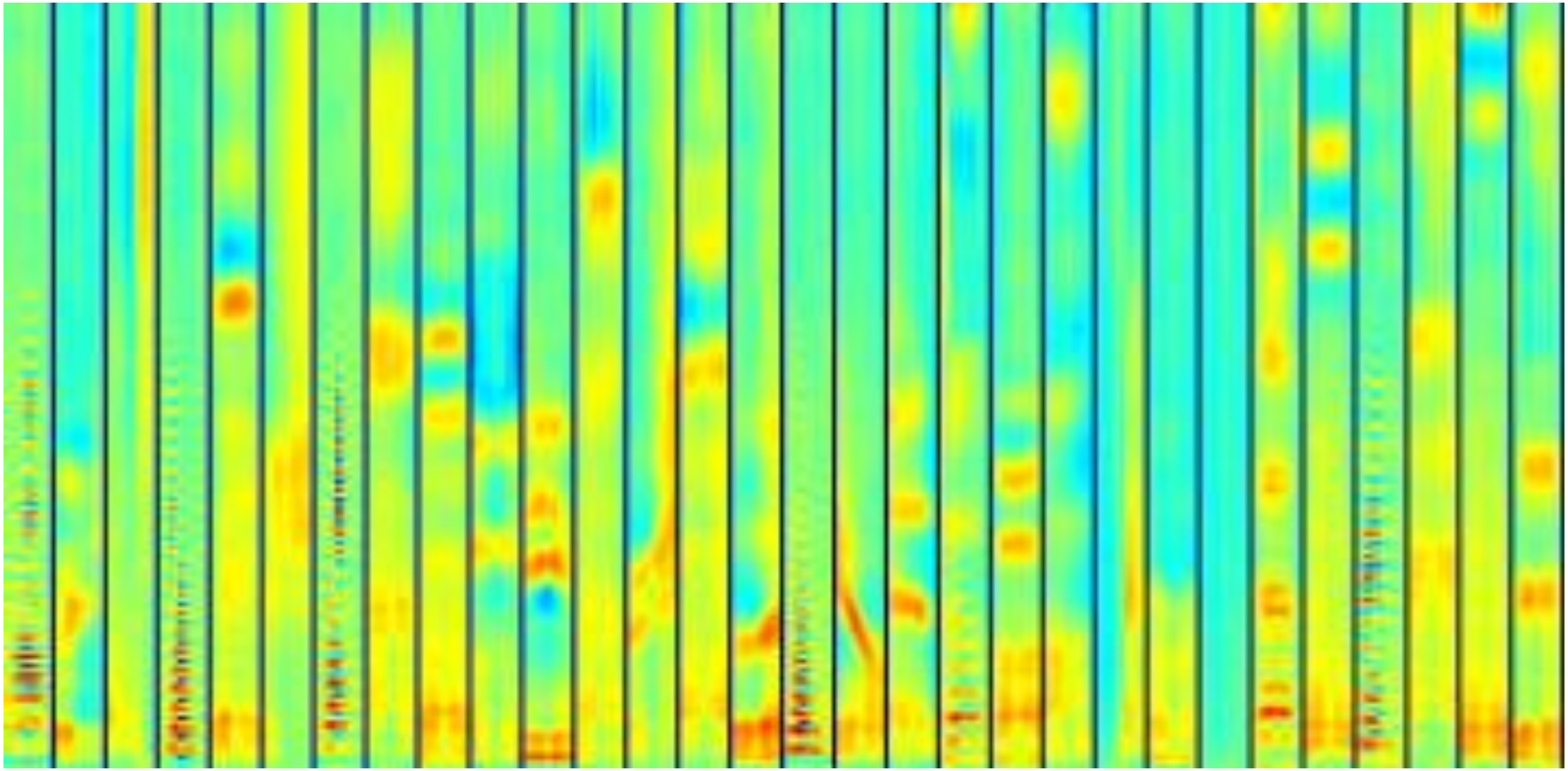
Convolutional DBN for audio



[Lee, Largman, Pham, Ng, NIPS 2009]

CDBNs for speech

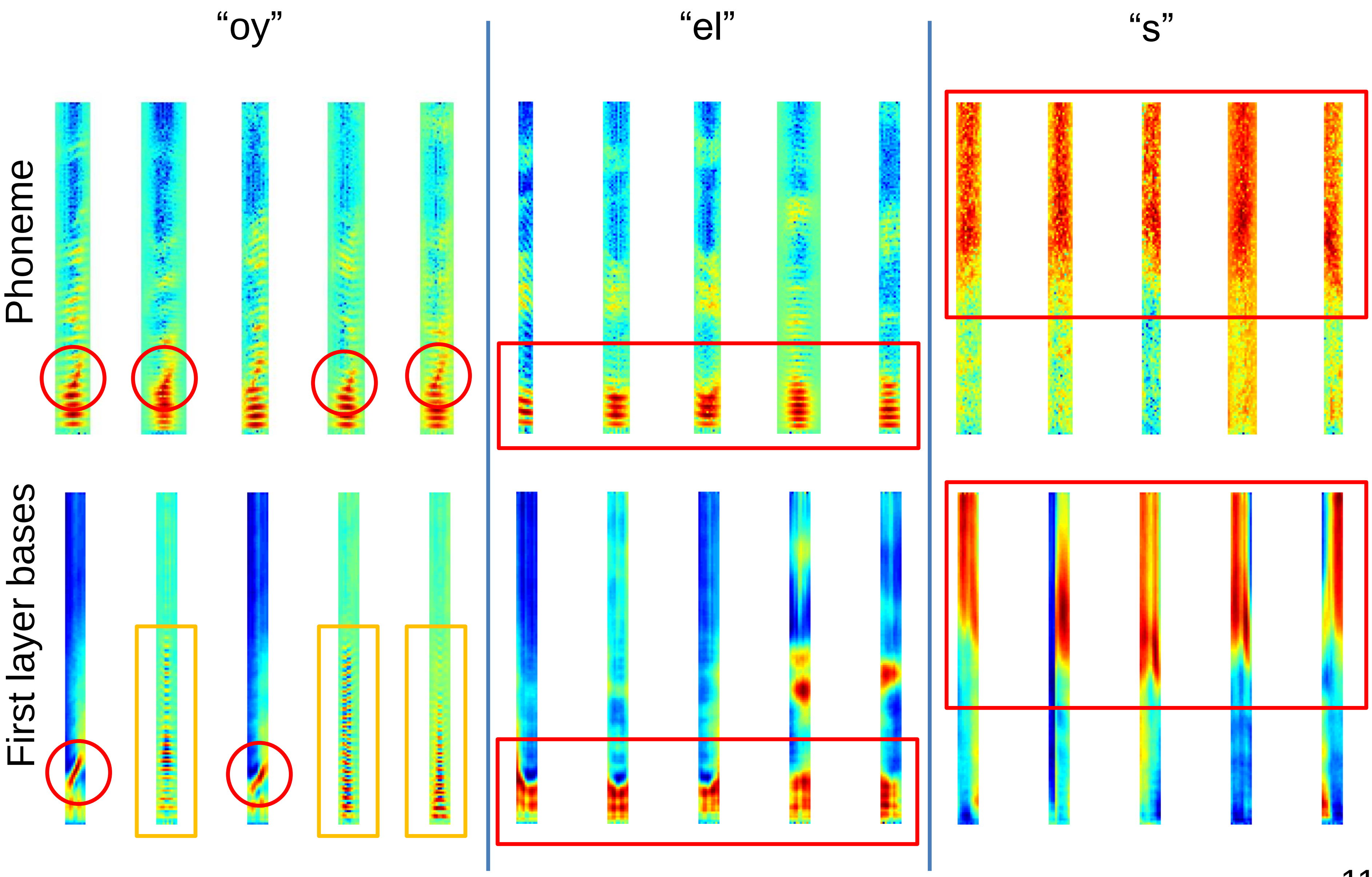
Trained on unlabeled TIMIT corpus



Learned first-layer bases

[Lee, Largman, Pham, Ng, NIPS 2009]

Comparison of bases to phonemes



Experimental Results

- Speaker identification

TIMIT Speaker identification	Accuracy
Prior art (Reynolds, 1995)	99.7%
Convolutional DBN	100.0%

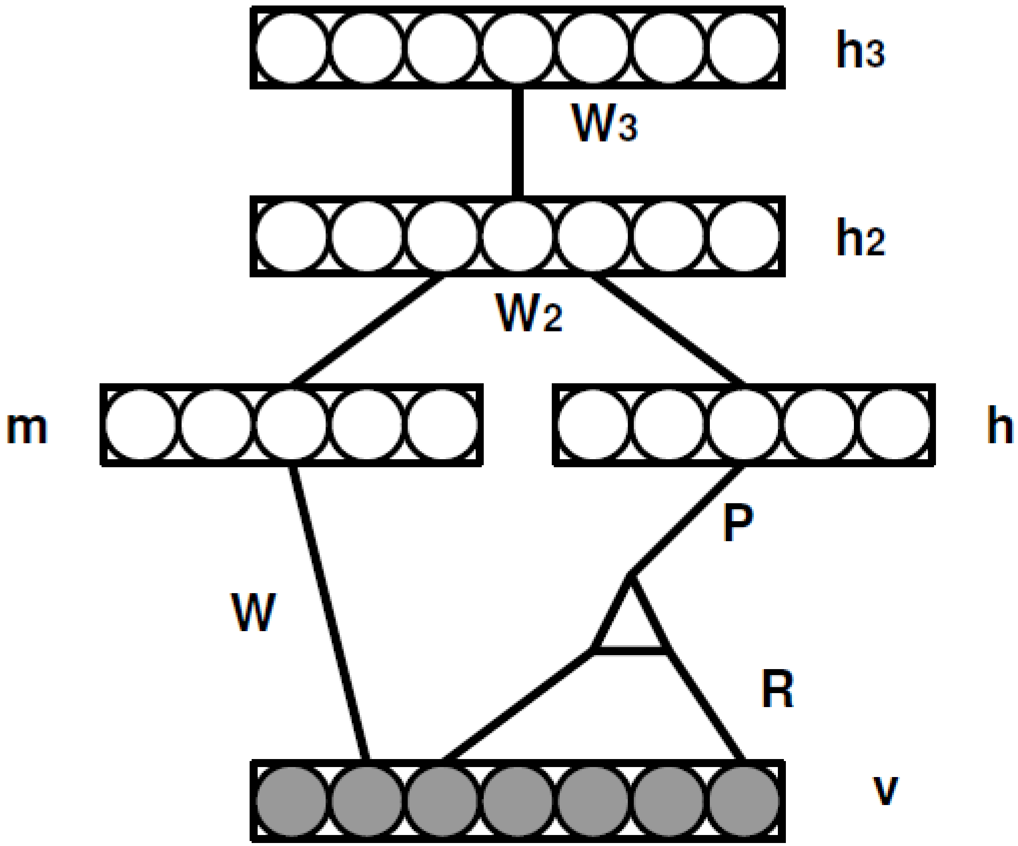
- Phone classification

TIMIT Phone classification	Accuracy
Clarkson et al. (1999)	77.6%
Petrov et al. (2007)	78.6%
Sha & Saul (2006)	78.9%
Yu et al. (2009)	79.2%
Convolutional DBN	80.3%
Transformation-invariant RBM (Sohn et al., ICML 2012)	81.5%

[Lee, Pham, Largman, Ng, NIPS 2009]

Phone recognition using mcRBM

- Mean-covariance RBM + DBN



Mean-covariance RBM

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{d}^{\mathrm{T}}\mathbf{h} - (\mathbf{v}^{\mathrm{T}}\mathbf{R})^2\mathbf{P}\mathbf{h},$$

[Dahl, Ranzato, Mohamed, Hinton, NIPS 2009]

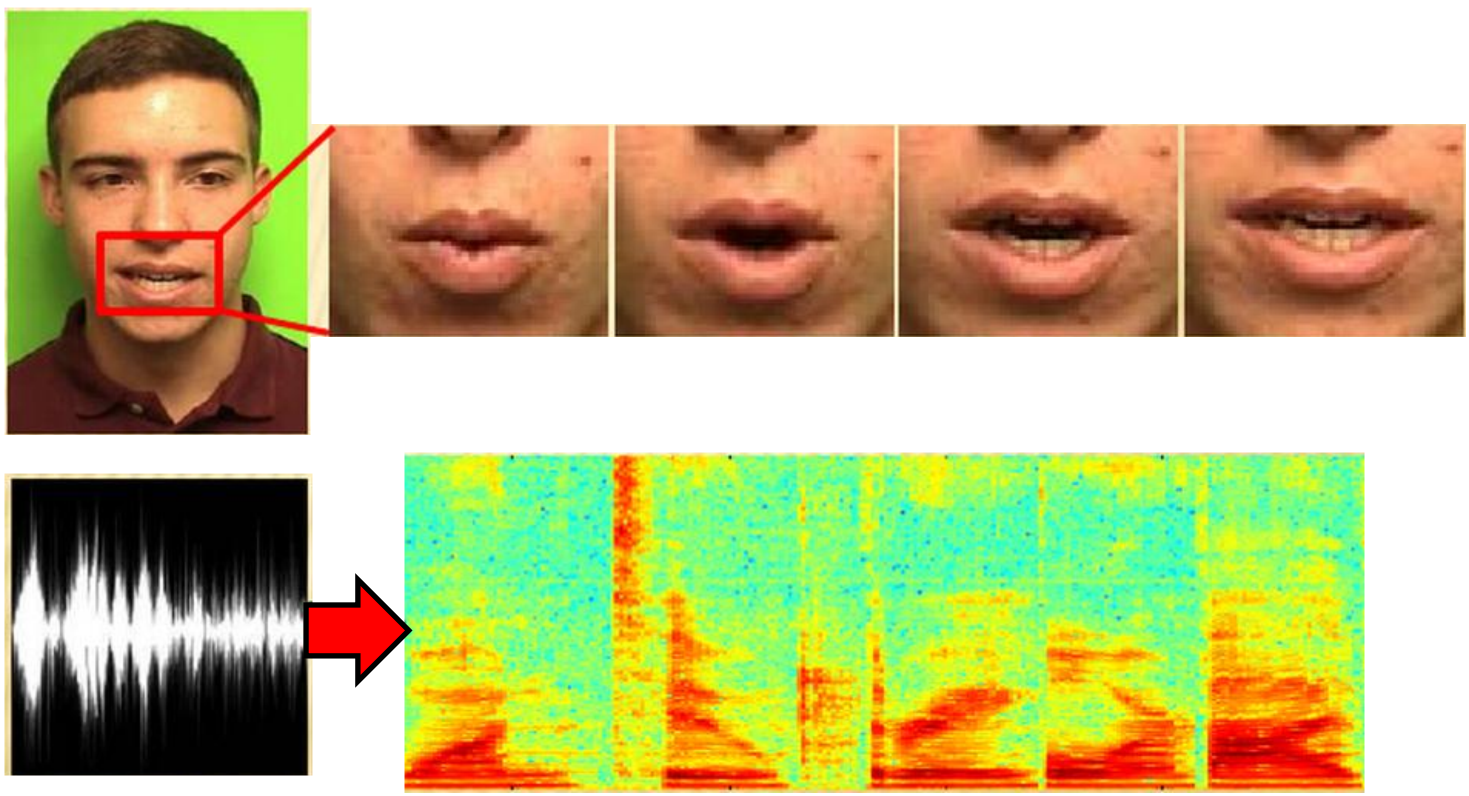
Speech Recognition on TIMIT

Method	PER
Stochastic Segmental Models	36.0%
Conditional Random Field	34.8%
Large-Margin GMM	33.0%
CD-HMM	27.3%
Augmented conditional Random Fields	26.6%
Recurrent Neural Nets	26.1%
Bayesian Triphone HMM	25.6%
Monophone HTMs	24.8%
Heterogeneous Classifiers	24.4%
Deep Belief Networks(DBNs)	**23.0%**
Triphone HMMs discriminatively trained w/ BMMI	22.7%
Deep Belief Networks with mcRBM feature extraction	**20.5%**

(Dahl et al., NIPS 2010)

Multimodal Feature Learning

- Lip reading via multimodal feature learning (audio / visual data)

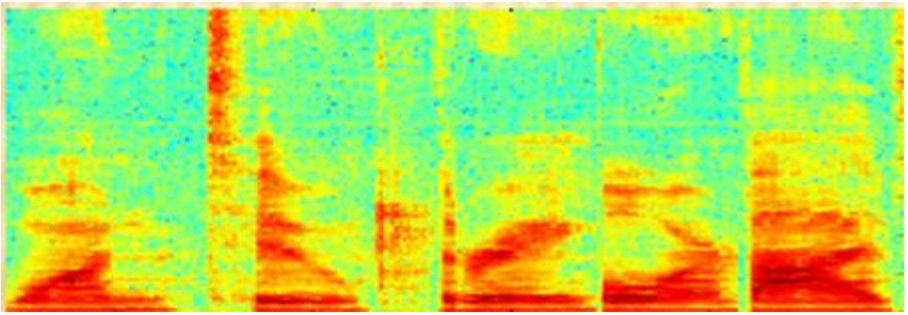


Slide credit: Jiquan Ngiam

Multimodal Feature Learning

- Lip reading via multimodal feature learning (audio / visual data)

$$\begin{bmatrix} 1.5 \\ -0.1 \\ 0 \\ 0.3 \\ . \\ . \\ . \\ 2.0 \end{bmatrix}$$



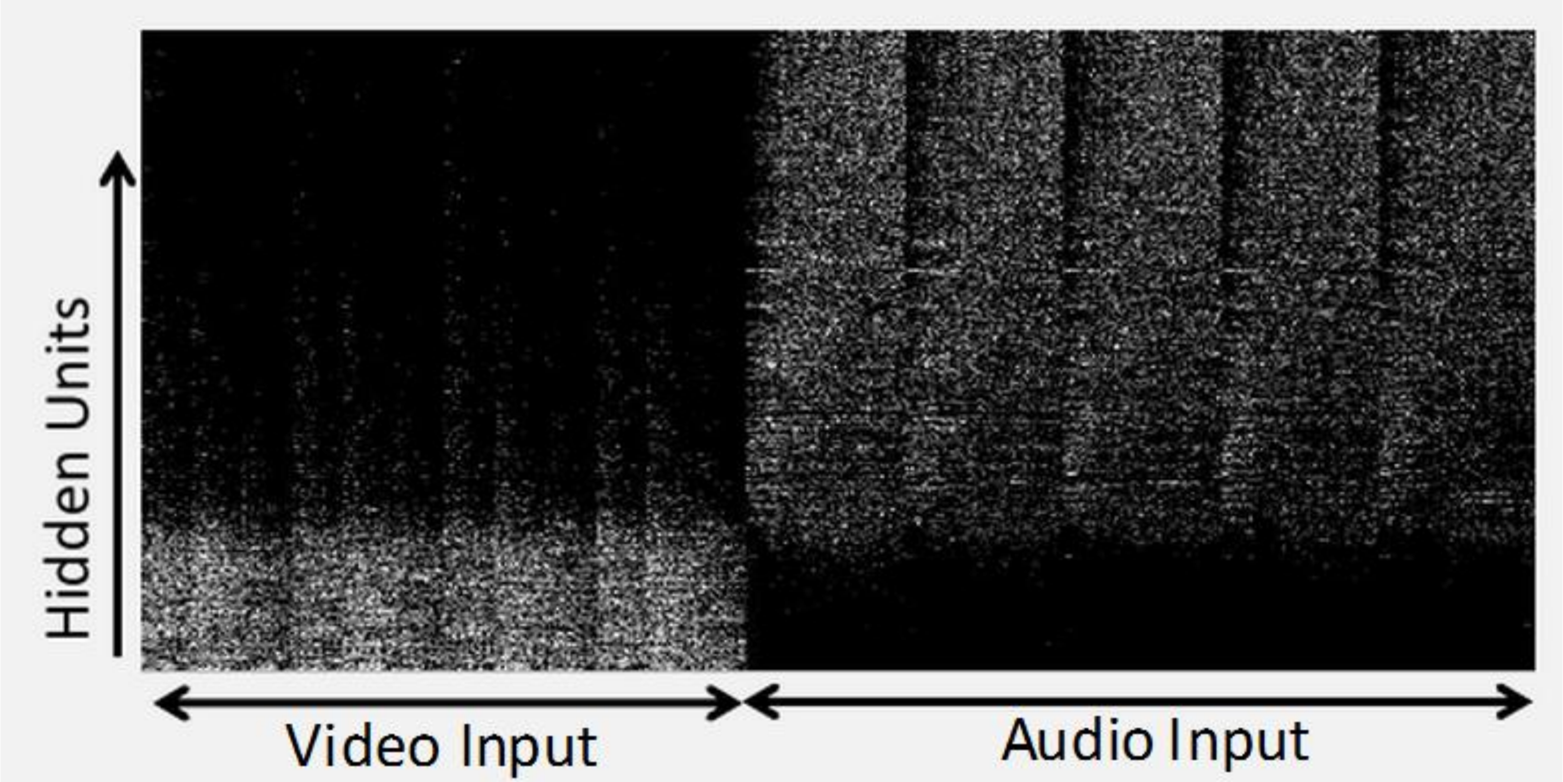
$$\begin{bmatrix} 0 \\ -0.3 \\ 0 \\ -0.8 \\ . \\ . \\ . \\ 1.1 \end{bmatrix}$$

Q. Is concatenating the best option?

Slide credit: Jiquan Ngiam

Multimodal Feature Learning

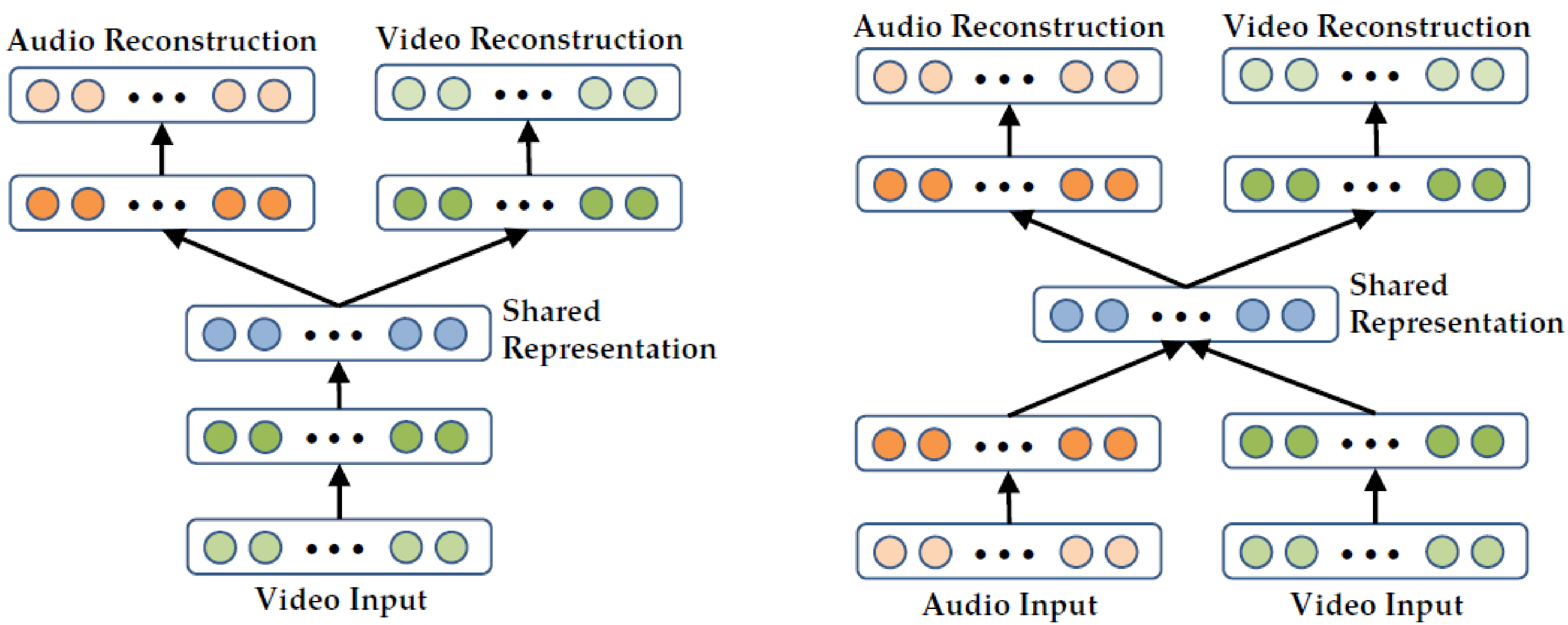
- Concatenating and learning features (via a single layer)doesn’t work



Mostly unimodal features are learned

Multimodal Feature Learning

- Bimodal autoencoder
  - Idea: predict unseen modality from observed modality

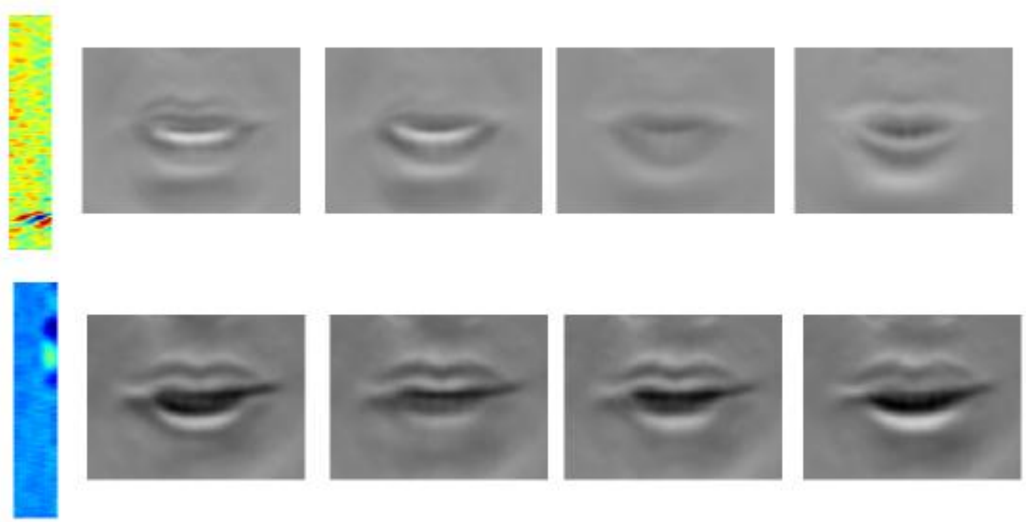


(a) Video-Only Deep Autoencoder (b) Bimodal Deep Autoencoder

Ngiam et al., ICML 2011

Multimodal Feature Learning

- Visualization of learned filters



Audio(spectrogram) and Video features learned over 100ms windows

- Results: AVLetters Lip reading dataset

Method	Accuracy
Prior art (Zhao et al., 2009)	58.9%
Multimodal deep autoencoder (Ngiam et al., 2011)	**65.8%**

Summary

- Learning Feature Representations
  - Restricted Boltzmann Machines
  - Deep Belief Networks
  - Stacked Denoising Autoencoders
- Deep learning algorithms and unsupervised feature learning algorithms show promising results in many applications
  - vision, audio, multimodal data, and others.

Thank you!