Latent variable models and “deep” learning

Chris Williams
School of Informatics, University of Edinburgh

March 2011

Motivations

- What lies beneath?

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables
- Learning a *representation* of the data; facilitates subsequent tasks

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables
- Learning a *representation* of the data; facilitates subsequent tasks
- Spearman (1904): Children's scores on a set of exams (e.g. Classics, French, English) might be explained by an underlying notion of general intelligence 'g'

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables
- Learning a *representation* of the data; facilitates subsequent tasks
- Spearman (1904): Children's scores on a set of exams (e.g. Classics, French, English) might be explained by an underlying notion of general intelligence 'g'
- Understand monitoring data from a jet engine in terms of faults, etc

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables
- Learning a *representation* of the data; facilitates subsequent tasks
- Spearman (1904): Children's scores on a set of exams (e.g. Classics, French, English) might be explained by an underlying notion of general intelligence 'g'
- Understand monitoring data from a jet engine in terms of faults, etc
- Frey and Jojic video

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables
- Learning a *representation* of the data; facilitates subsequent tasks
- Spearman (1904): Children's scores on a set of exams (e.g. Classics, French, English) might be explained by an underlying notion of general intelligence 'g'
- Understand monitoring data from a jet engine in terms of faults, etc
- Frey and Jojic video
- "We are drowning in information, but starving for knowledge!" (Naisbett, 1982)

Motivations

- What lies beneath?
- The surface structure of the data can be best explained in terms of some underlying, hidden variables
- Learning a *representation* of the data; facilitates subsequent tasks
- Spearman (1904): Children's scores on a set of exams (e.g. Classics, French, English) might be explained by an underlying notion of general intelligence 'g'
- Understand monitoring data from a jet engine in terms of faults, etc
- Frey and Jojic video
- "We are drowning in information, but starving for knowledge!" (Naisbett, 1982)
- These are also some of the problems faced by the brain

Outline

1. A crash course in graphical models
2. Models with one layer of hidden variables
3. Modelling sequences
4. Going deep
5. Discussion

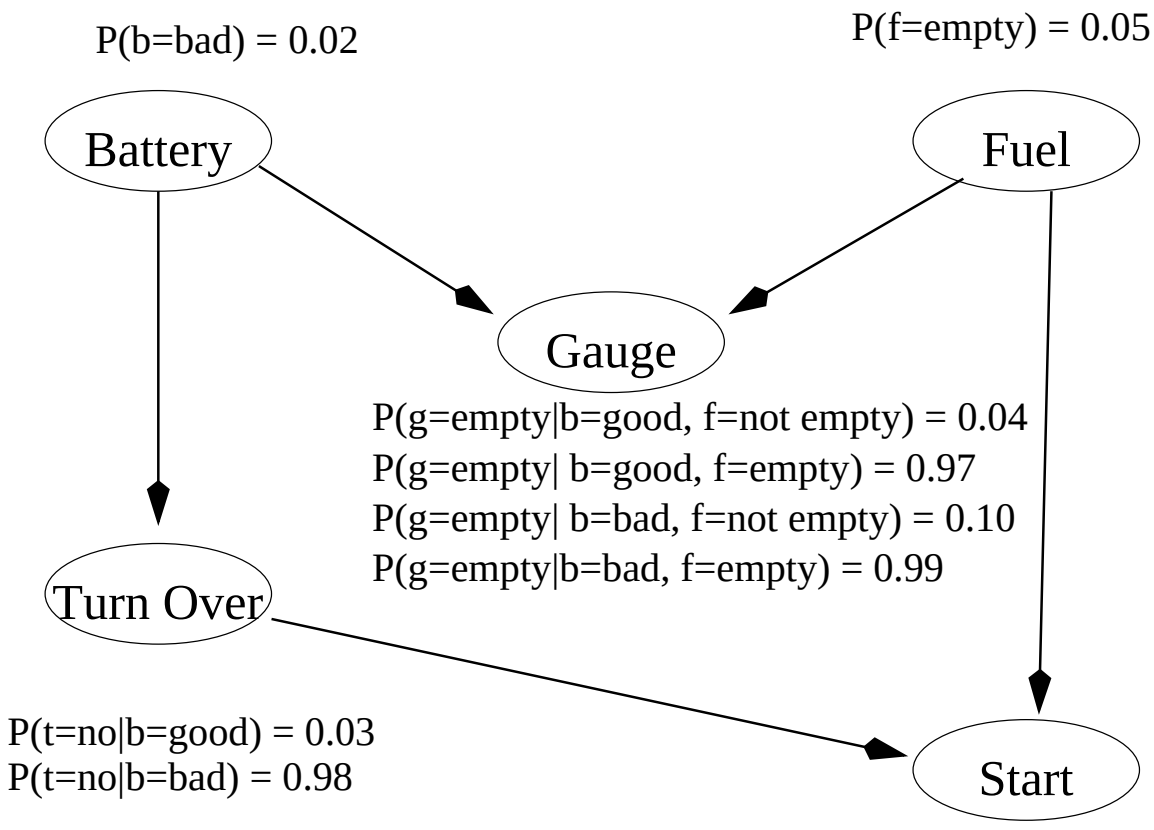
1. Probabilistic graphical models

- Probabilistic graphical models are a tool for modelling complex networks of relationships which are non-deterministic
- In a directed graphical model G (aka Bayesian network) the joint probability is defined as

$$p(X_1, \ldots, X_m | G) = \prod_i p(X_i | \text{parents}_i, G)$$

- It is the *missing* edges that describe conditional independences
- Model is comprised of *structure* and *parameters*
- Bayesian networks represent conditional (in)dependence relations: not necessarily causal interactions

Example: Does my car start?



Heckerman (1995)

$$p(b, f, g, t, s) = p(b)p(f)p(g|b, f)p(t|b)p(s|t, f)$$

Undirected graphical models

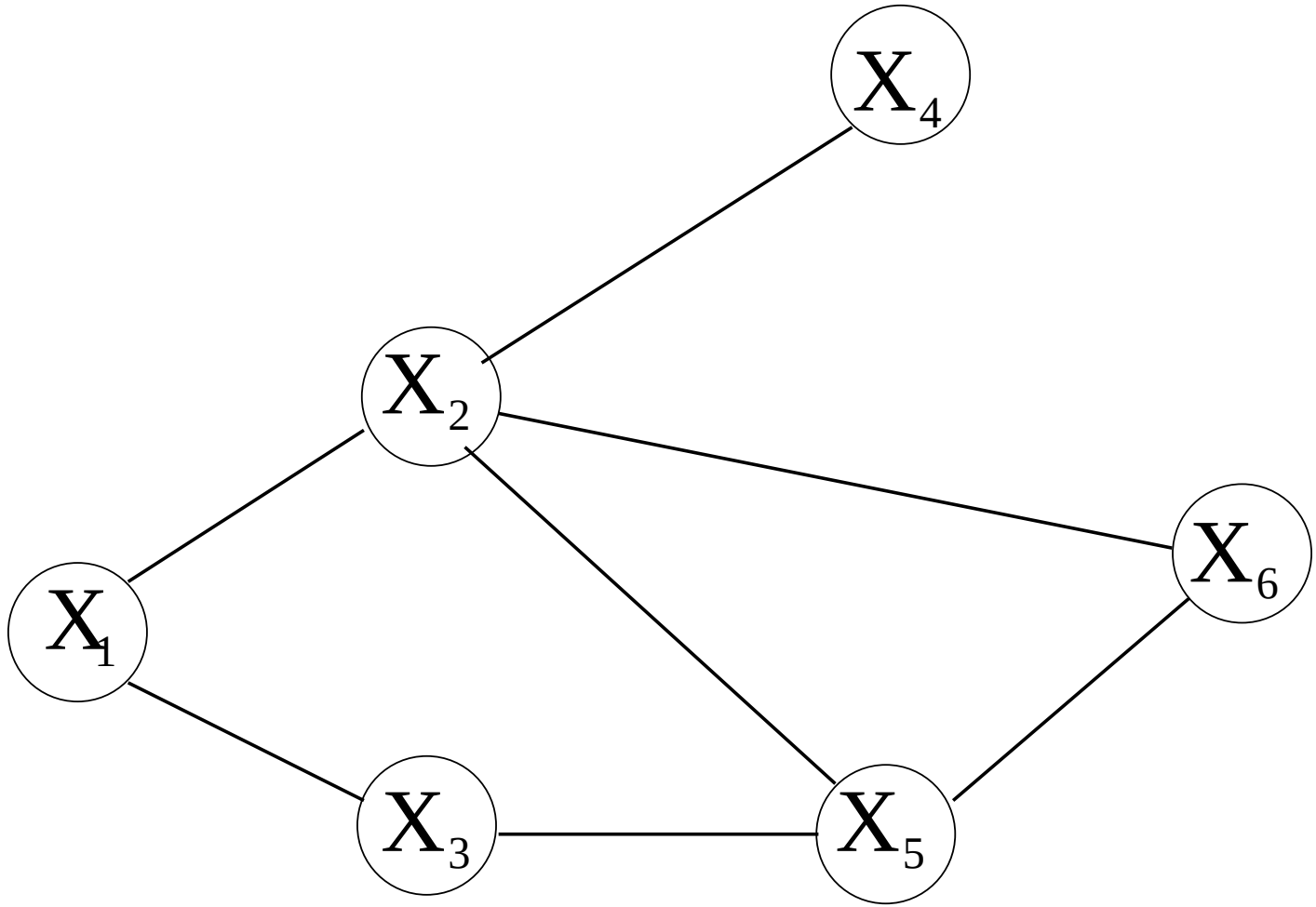
- For undirected graphs, locality depends on the notion of cliques
- Joint probability distribution is given as a product of local functions defined on the maximal cliques of the graph

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_{X_C}(\mathbf{x}_C)$$

with

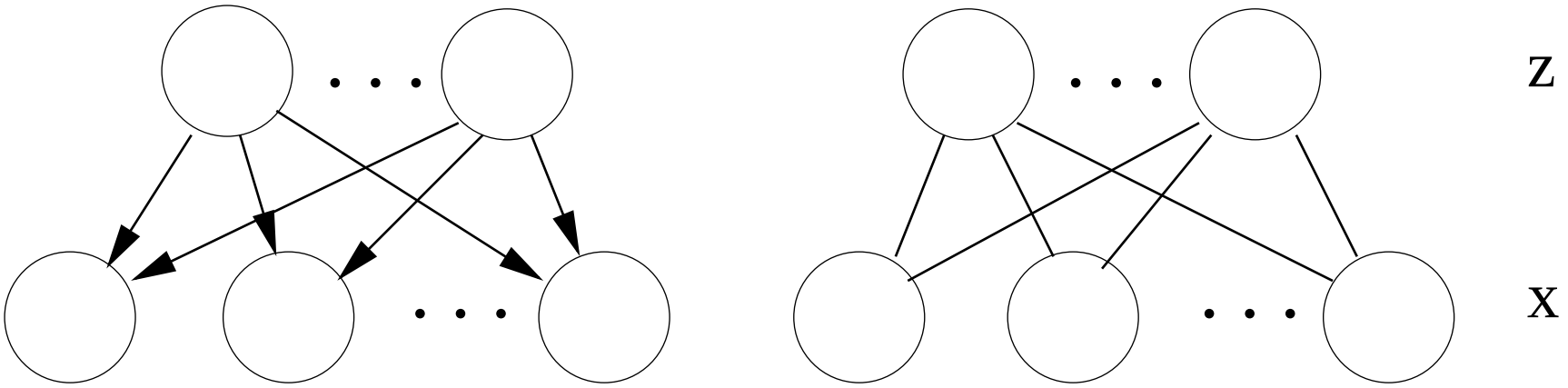
$$Z = \sum_{\mathbf{x}} \prod_{C \in \mathcal{C}} \psi_{X_C}(\mathbf{x}_C)$$

- Each $\psi_{X_C}(\mathbf{x}_C)$ is a strictly positive, real-valued function, otherwise arbitrary
- Z is called the partition function



$$p(\mathbf{x}) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_3, x_5)\psi(x_2, x_5, x_6)\psi(x_2, x_4)$$

2. Models with one layer of hidden variables



directed undirected

- The directed model is the most common formulation, but we will see the undirected model used later for “deep learning”
- Hidden and visible variables can be continuous, discrete etc
- Examples: clustering, factor analysis, topic models (LDA), sparse coding, multiple cause vector quantization (MCVQ)

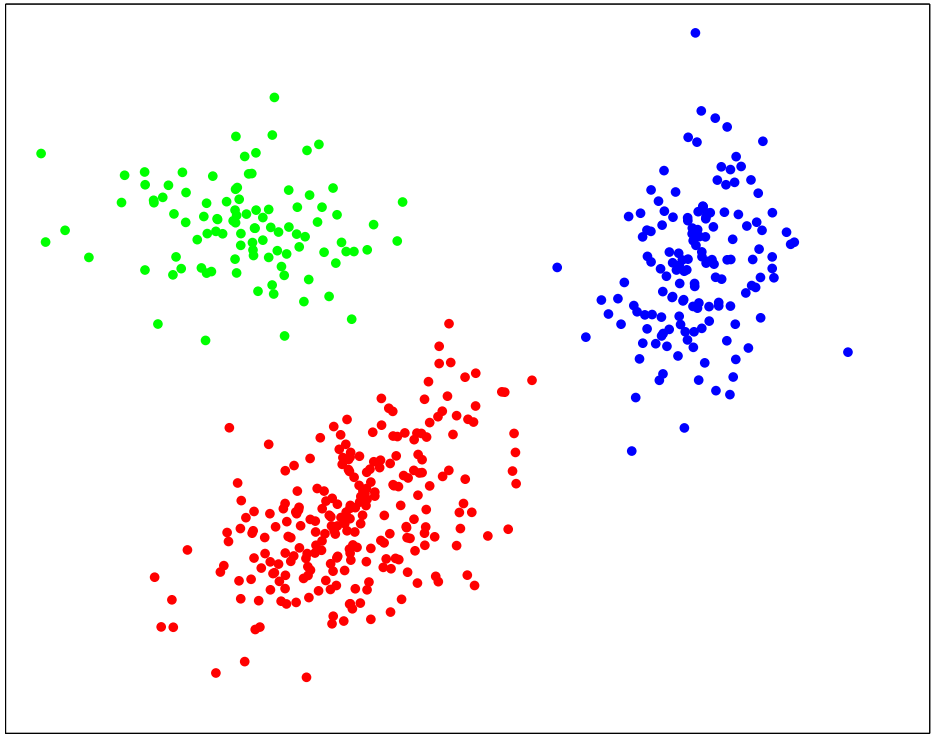
What are latent variables?

- We have an observed data vector $\mathbf{x}$
- We assume that the inter-relationships between the observed (or manifest) variables in $\mathbf{x}$ can be explained in terms of some latent (or hidden) variables $\mathbf{z}$

$$p(\mathbf{x}) = \int_{\mathbf{z}} \prod_i p(x_i|\mathbf{z}) p(\mathbf{z}) \, d\mathbf{z}$$

- Reasons for being hidden: latent variables may be theoretical concepts, or real physical variables that are simply unobserved (e.g. the location of an object in a scene)
- Caveat: inference of *causality* is a hard problem. Generally needs experiments (actions) as well as observational data

Clustering/Mixture Models



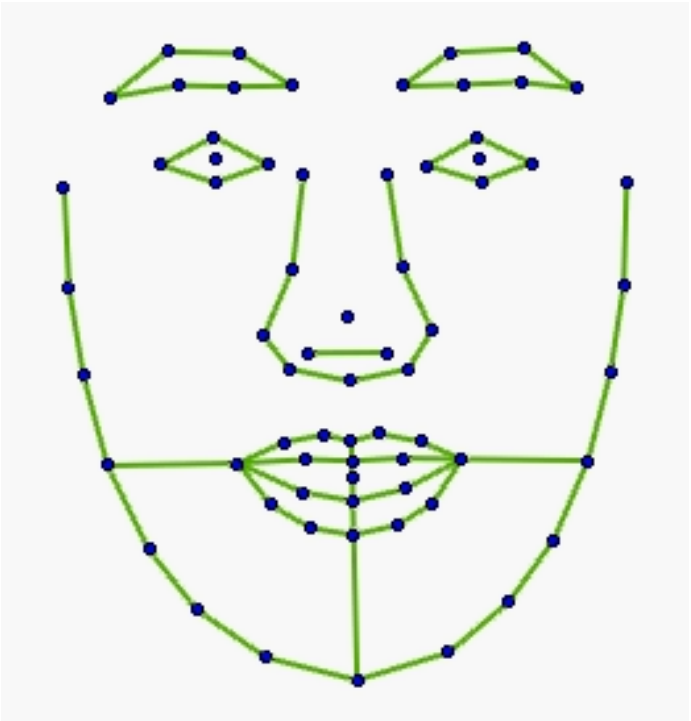
- z has a one-on-in-k coding
- Examples: Clustering of documents; discovery of new classes of infra-red stars in the IRAS Low Resolution Spectral catalogue (Goebel et al, 1989)
- Clustering is limited, we want multiple causes

Factor analysis

- *Factorized* Gaussian model for $\mathbf{z}$ variables
- Example: Data for each frame is (x, y) locations of many markers on a face concatenated into a vector
- Model rigid and non-rigid motion in terms of a small number of modes of deformation $\mathbf{w}_i$

$$\mathbf{x} = \sum_i z_i \, \mathbf{w}_i + \text{noise}$$

- Tim Cootes' animations



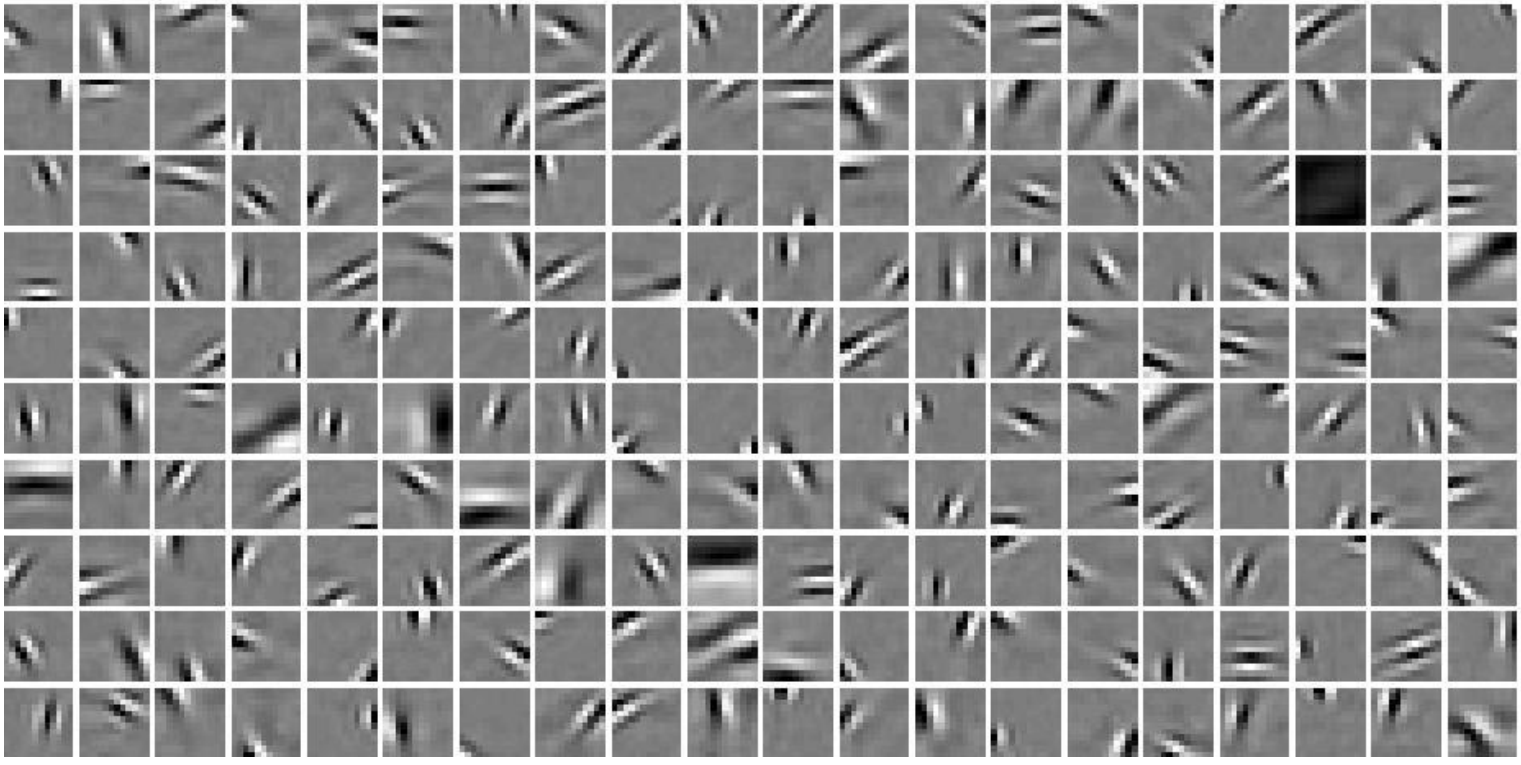
Sparse Coding

Olshausen and Field (1996, 1997)

- Model patches of natural images $\mathbf{x}$ as a superposition of basis functions $\mathbf{w}_i$ with strengths z_i

$$\mathbf{x} = \sum_i z_i \mathbf{w}_i + \text{noise}$$

- Model is *overcomplete*, so there are more basis functions than pixels
- *Sparsity prior* so that z's are mostly near zero and only occasionally large



Overcomplete: 200 basis functions from 12×12 patches

[Figure: Olshausen, 2005]

Topic Modelling

Blei et al (2003)

- Bag-of-words representation for each document (ignore word order)
- Each document is regarded as being generated from a weighted set of topics
- Each topic is a probability distribution over words
- $0 \leq z_i \leq 1$ for all i and $\sum_i z_i = 1$
- Learn topics and estimate per-document topic weightings
- AP corpus, 16,333 articles with 23,075 unique terms

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

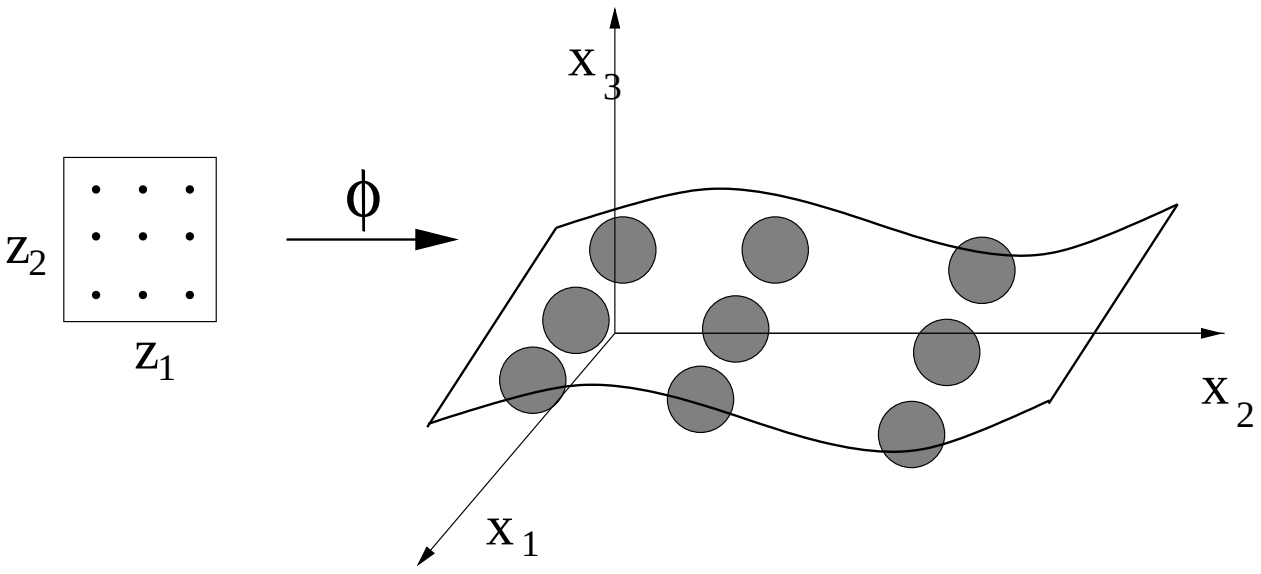
Learning Latent Variable Models

- Use the Expectation-Maximization algorithm (Dempster, Laird and Rubin, 1977)
- Goal is to find parameters θ that maximize the log likelihood

$$L = \sum_{i=1}^{n} \log p(\mathbf{x}_i|\theta) = \sum_{i=1}^{n} \log \sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i|\theta)$$

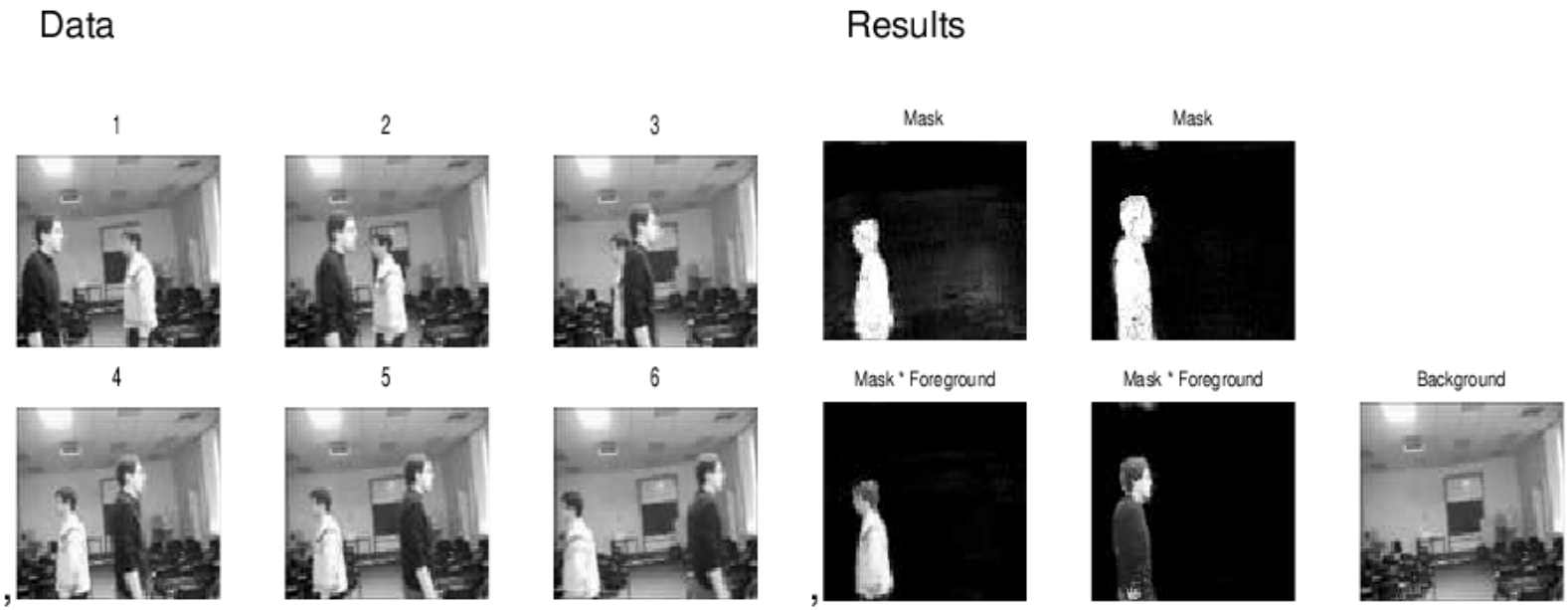
- Do this by iterating E- and M-steps
  - E step: Compute $p(\mathbf{z}_i|\mathbf{x}_i)$ for all data points $i = 1, \ldots, n$
  - M-step: Adjust the parameters to maximize (or at least increase) the expected complete data log likelihood
- This can be shown to converge to a local maximum of L
- In some cases approximations may be needed

Handling Non-linearity



- Generative Topographic Mapping (Bishop, Svensen and Williams, 1997/8)
- Difficulty with parameterizing the non-linear mapping; curse of dimensionality wrt the latent space dimension
- Gaussian Process Latent Variable Model (Lawrence 2005)
- More specific structure of the non-linearity

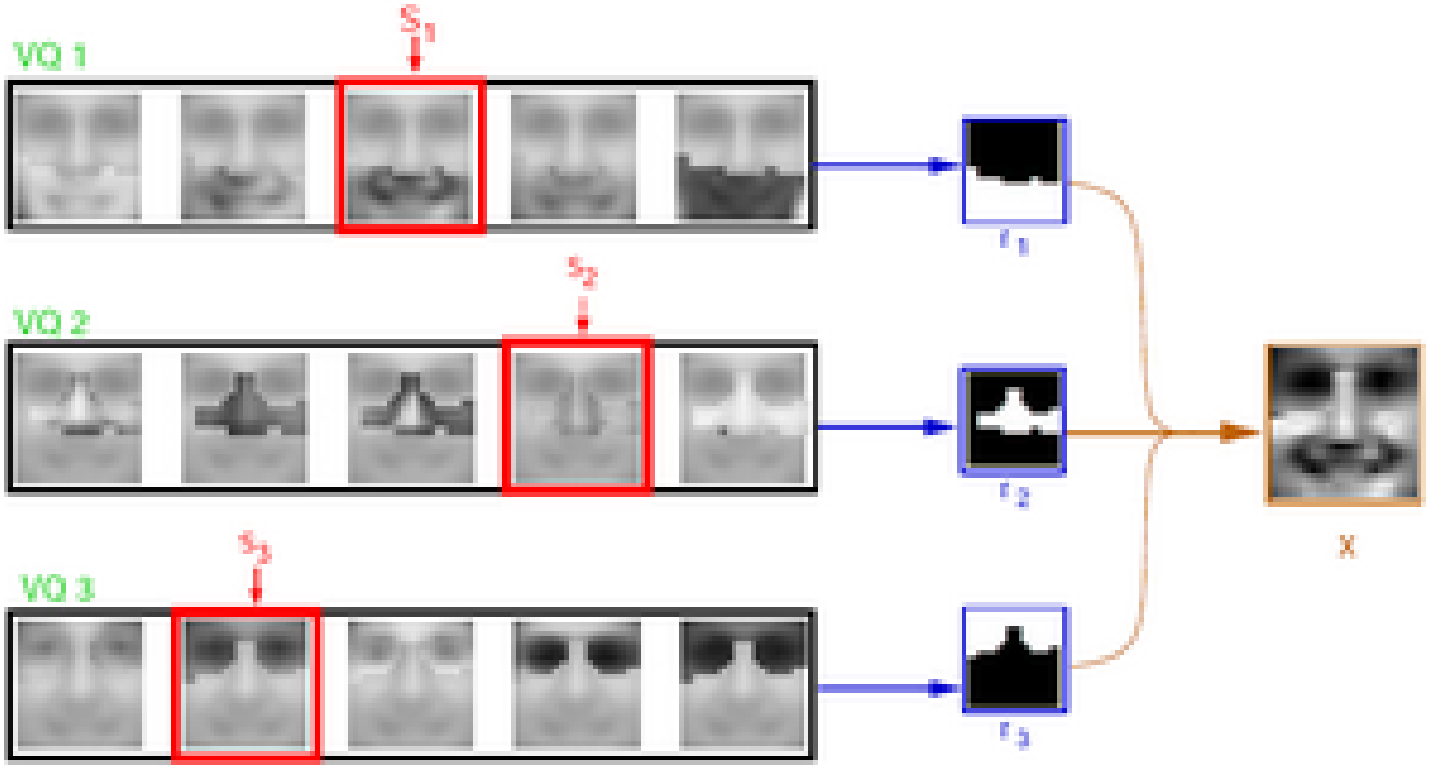
Williams and Titsias (2004)



- Uses greedy learning to extract one object at a time
- The latent variable for each object is its location
- Masks and depth ordering determine how layers combine

Multiple Cause Vector Quantization

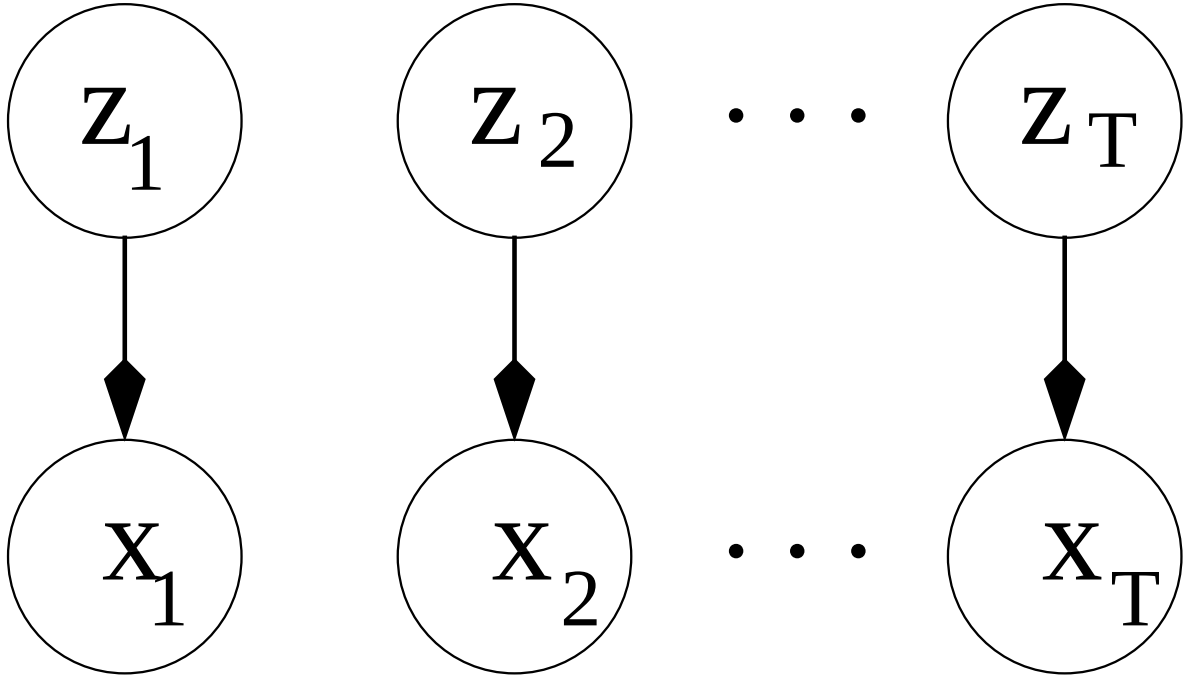
Ross and Zemel (2006)

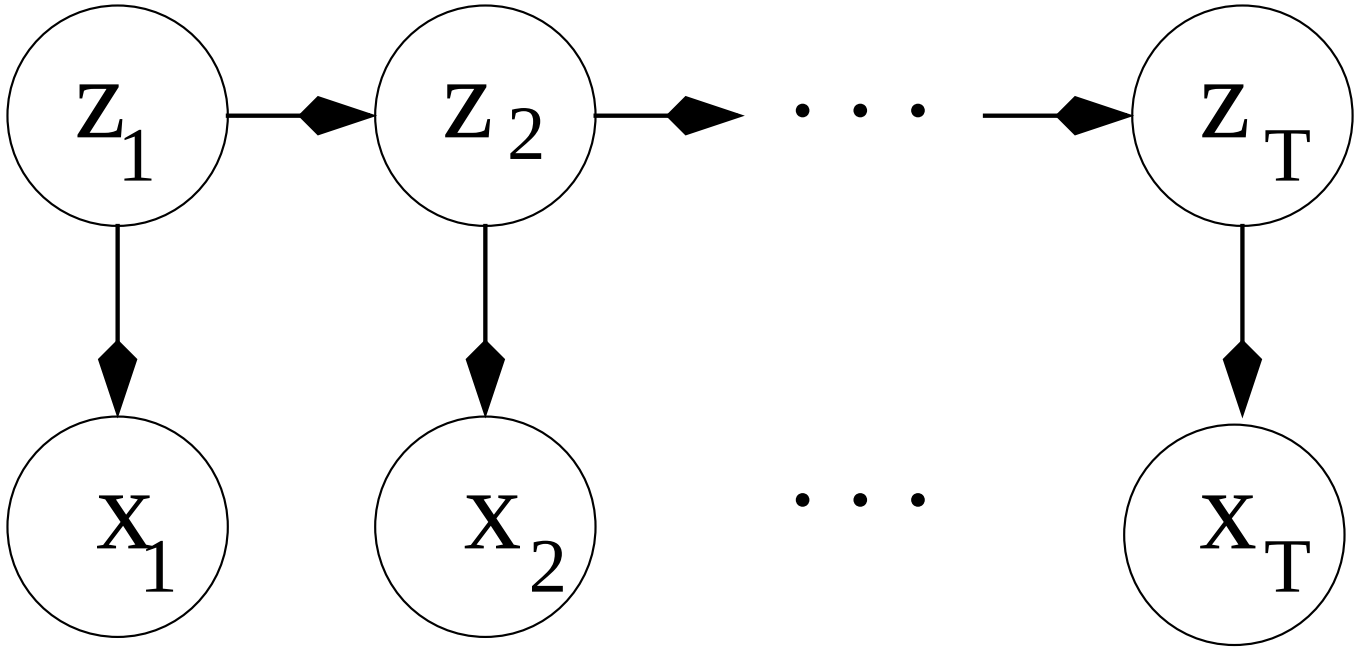


- Factorial face image generator
- The components are combined using a *mask*

3. Modelling sequences

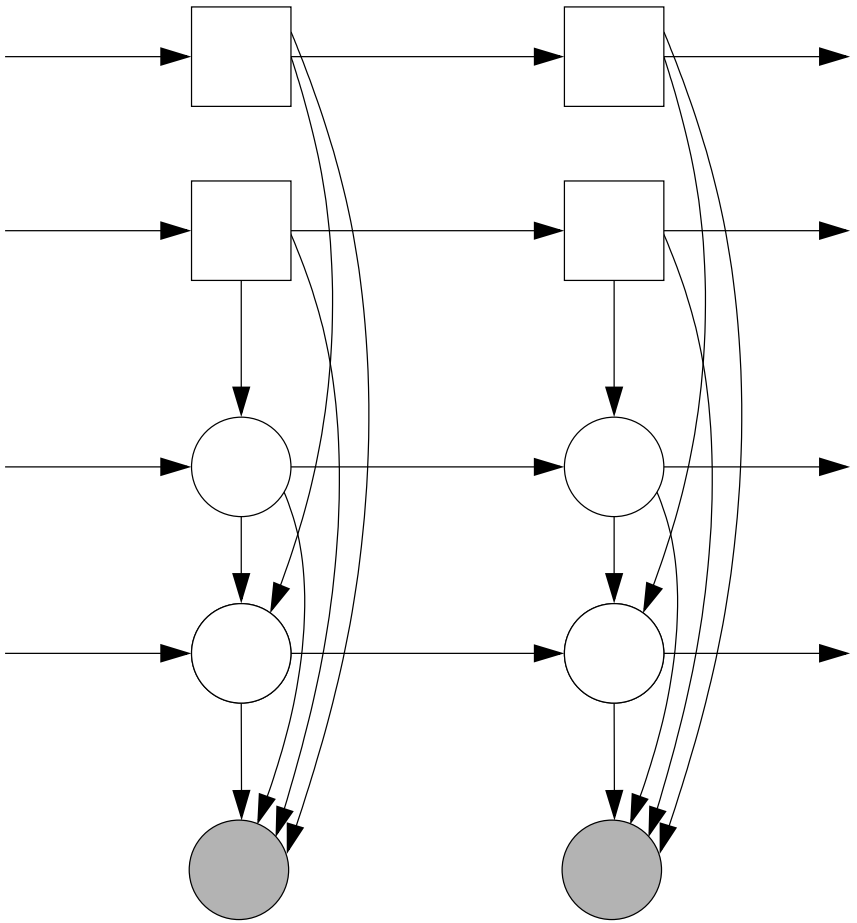
- A compressed representation of the one-layer latent variable model
- The figure shows T independent draws from the model





- Clustering $\rightarrow$ Hidden Markov model (HMM)
- Factor analysis $\rightarrow$ Kalman filter
- MCVQ $\rightarrow$ Factorial Hidden Markov model (FHMM)

Factorial Switching Kalman Filter



Artifactual factors

Physiological factors

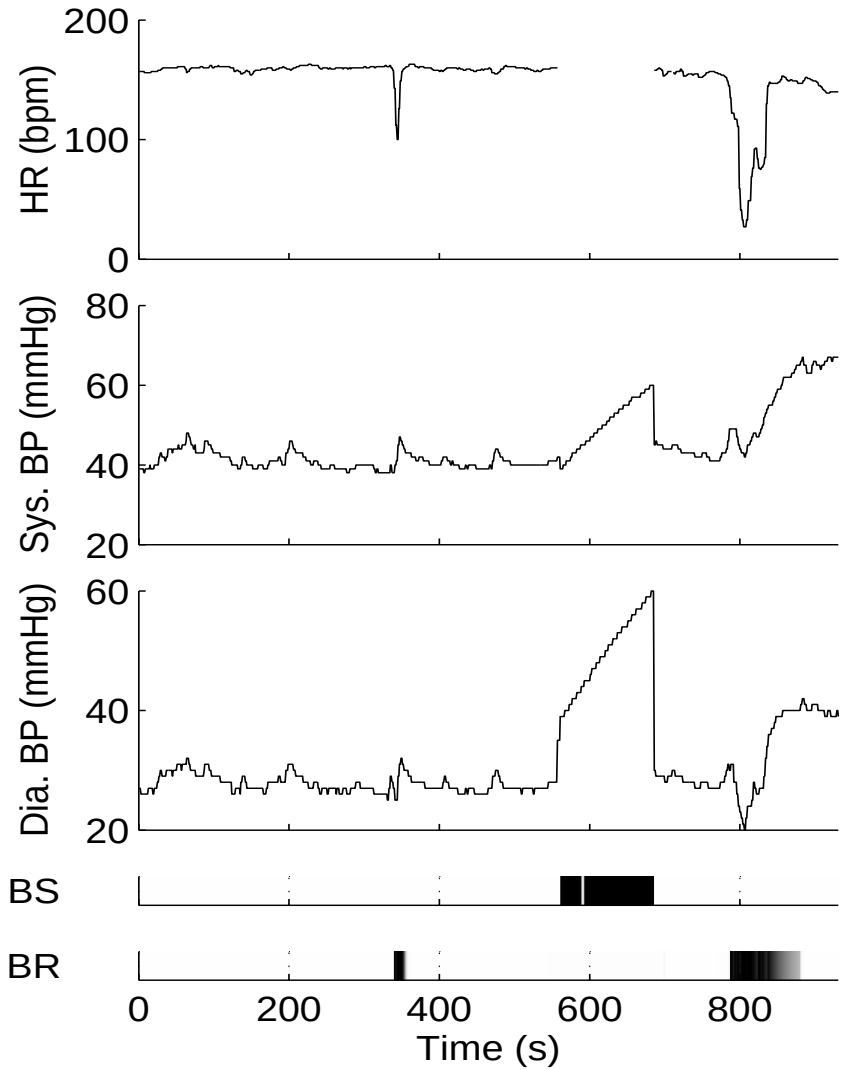
Physiological state

Artifactual state

Observations

Neonatal Condition Monitoring

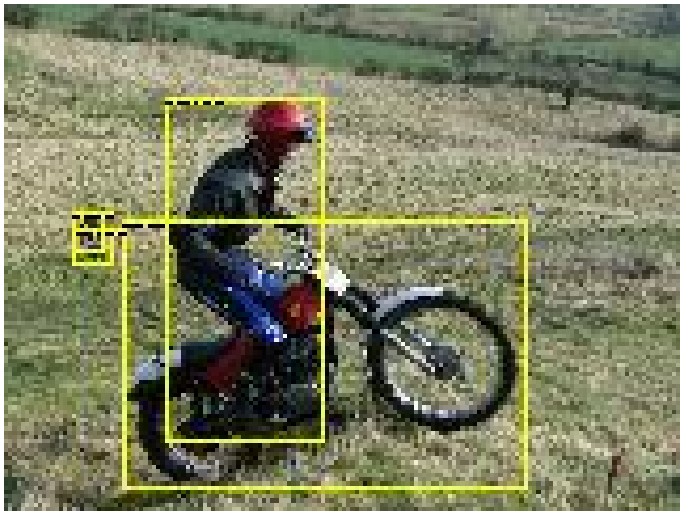
- Quinn, Williams and McIntosh (2009)
- Factors typically combine by overwriting (occlusion)

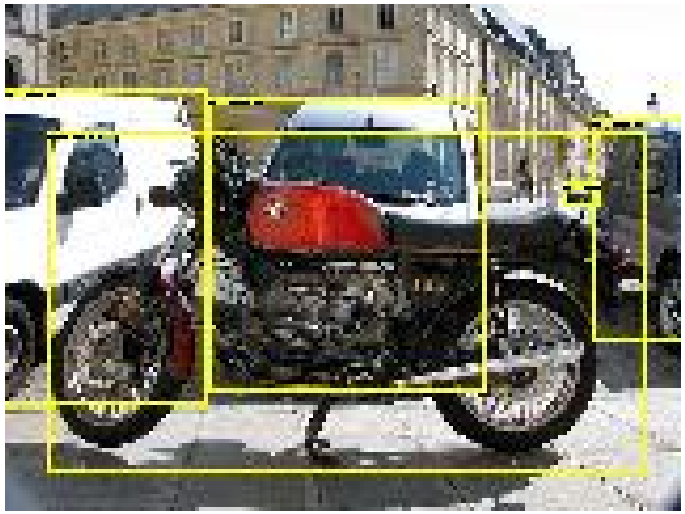


4. Beyond the single hidden layer: Hierarchy

- Hierarchy is a recurring theme in many areas of AI
- Many types of hierarchical structure, e.g.
  - PART-OF
  - AND-OR
  - IS-A
  - spatial, temporal scale

Hierarchical Structure in Scene Analysis





- Scene type e.g. indoor, outdoor rural, outdoor urban
- Objects occurring (and co-occurring) depend on scene type
- Inter-relationships between the pose of objects
- Objects can be composed of *parts*, e.g. car wheels, body, roof, with variable shape and appearance

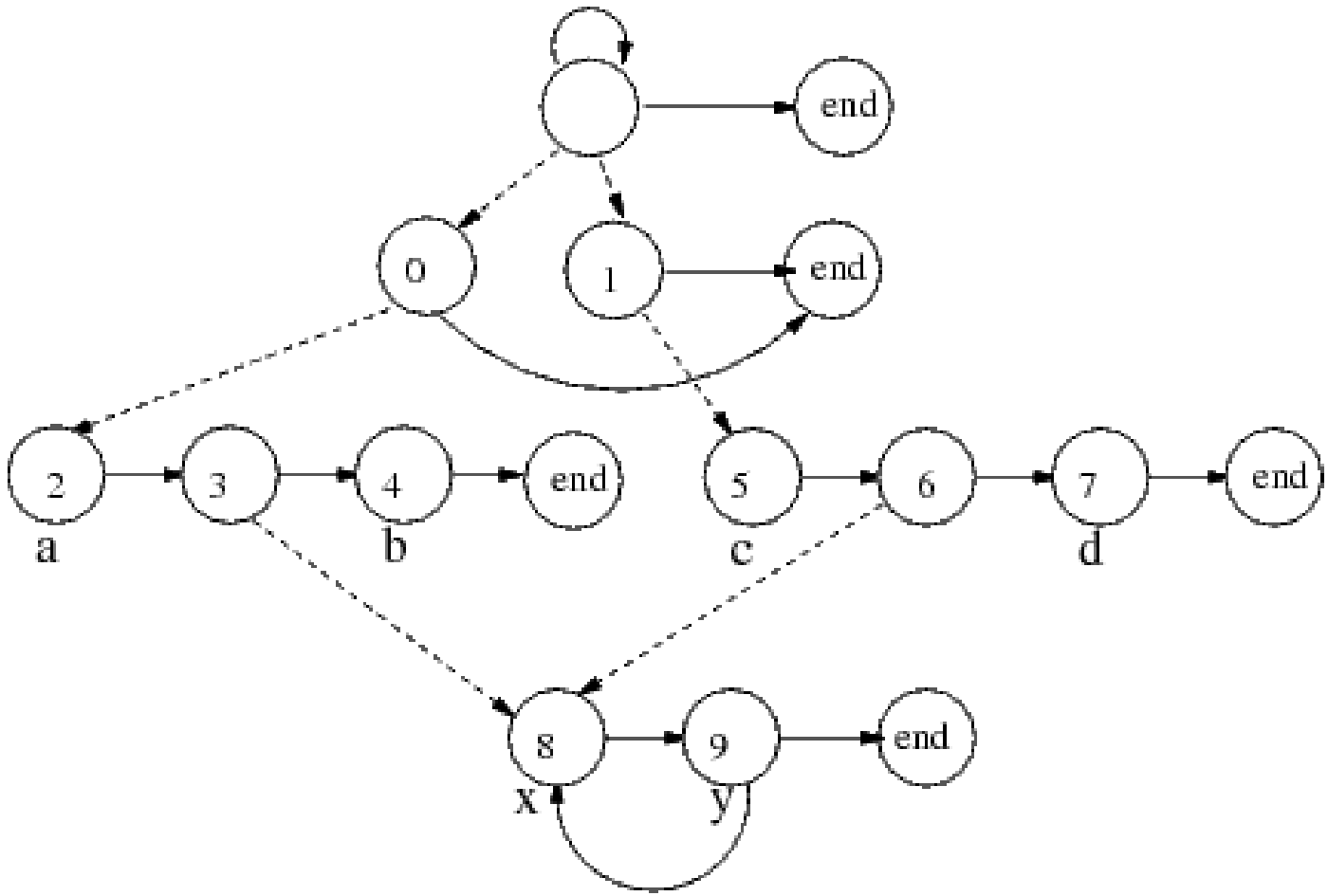
Other factors

- Depth, occlusions
- Illumination, shadows

Hierarchical Structure in Data Streams

- Consider cooking by following a recipe: there are a number of tasks, each of which may have sub-components
- The tasks will have some dependencies, but there can be inter-leaving of multiple tasks
- Animal behaviour modelling, e.g. a pair of Drosophila. High level states (anxiety, arousal, aggression) affect lower-level primitives, e.g. wing extension, walking, pirouetting, orienting, singing
- Analysing sports action
- Hierarchical HMMs (Fine, Singer & Tishby, 1998)

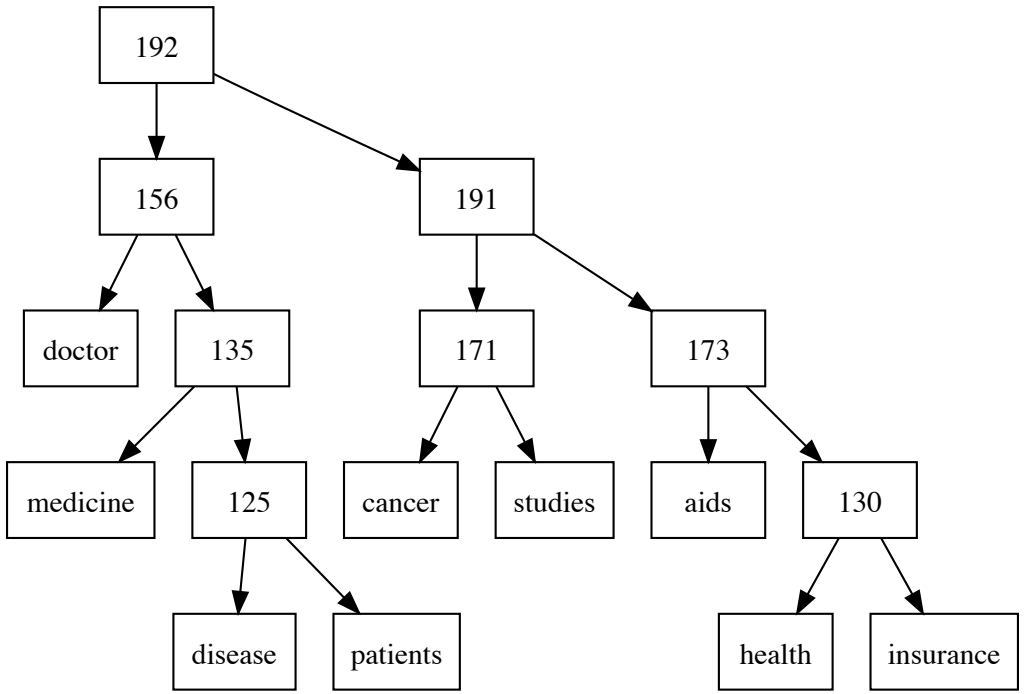
Hierarchical HMM as a Finite State Automaton



HFSA for $a(xy)^+b|c(xy)^+d$

Figure credit: Kevin Murphy (2002)

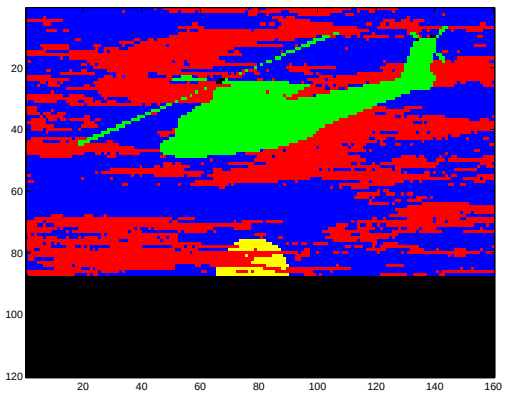
Trees

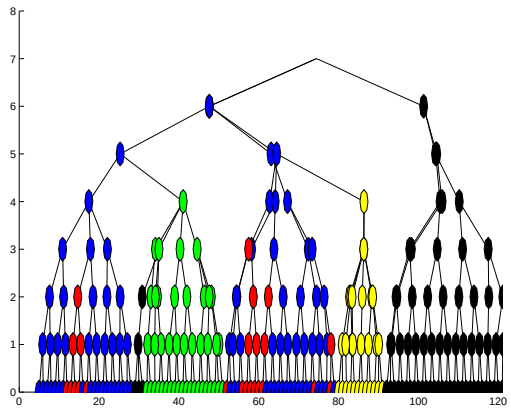


- Phylogenetic trees (Wright, 1921)
- Example: Harmeling and Williams (2010): building trees bottom up from data
- Grammars

Dynamic Trees

Storkey and Williams (2003)

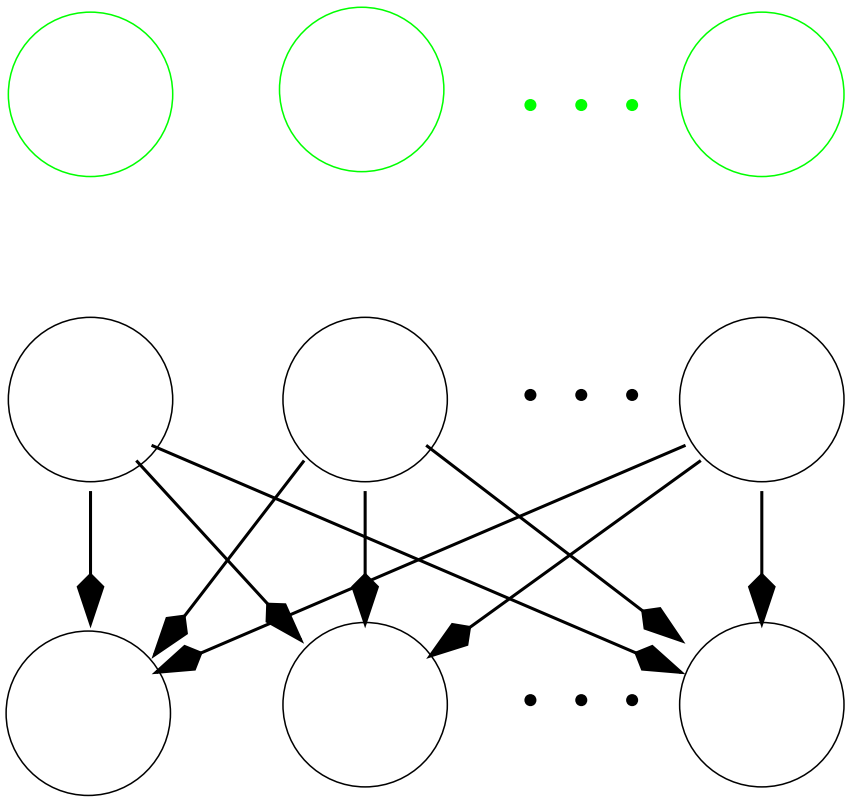




- Model builds a “parse tree” corresponding to each input image
- Parent-child relationships encoded were very simple in this model
- “Tower of jelly” problem

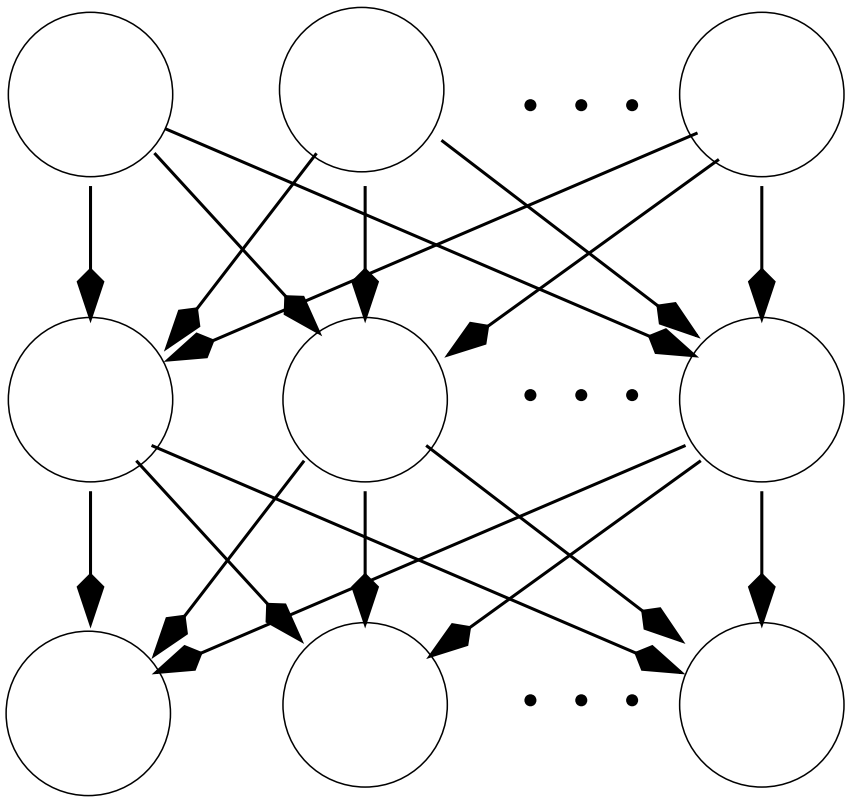
Beyond Trees

- Hierarchical Structure and Multiple Causes



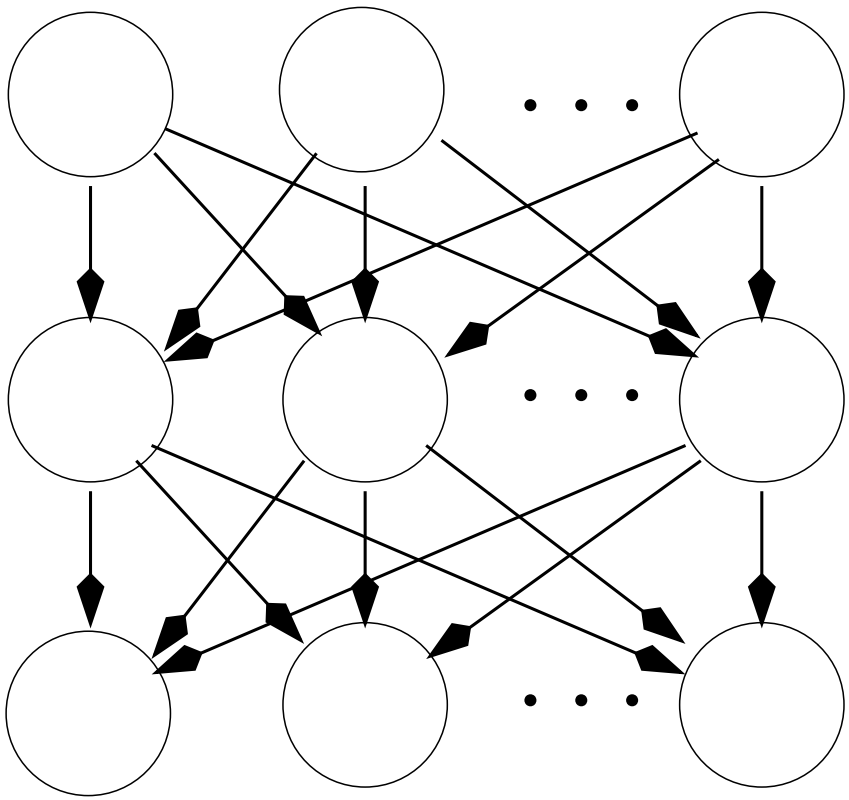
Beyond Trees

- Hierarchical Structure and Multiple Causes



Beyond Trees

- Hierarchical Structure and Multiple Causes



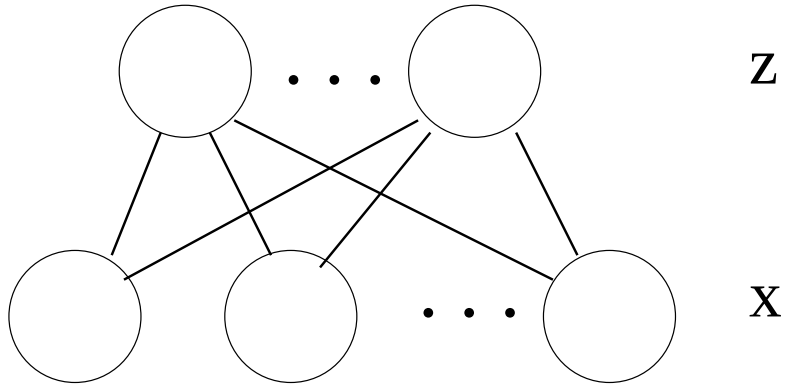
- “anything you can do, I can do meta”

Greedy Learning

- Start by learning a one-hidden-layer model
- The model's "independent causes" turn out to not really be independent (and we can see this by looking at the aggregated posterior over examples)
- Add another layer of units to model these correlations (needs to be non-linear)
- Possibly adjust the parameters in the lower layers once the higher layers have been learned
- Examples: Hyvärinen et al (2000, 2001), Karklin and Lewicki (2003, 2005); models of image patches

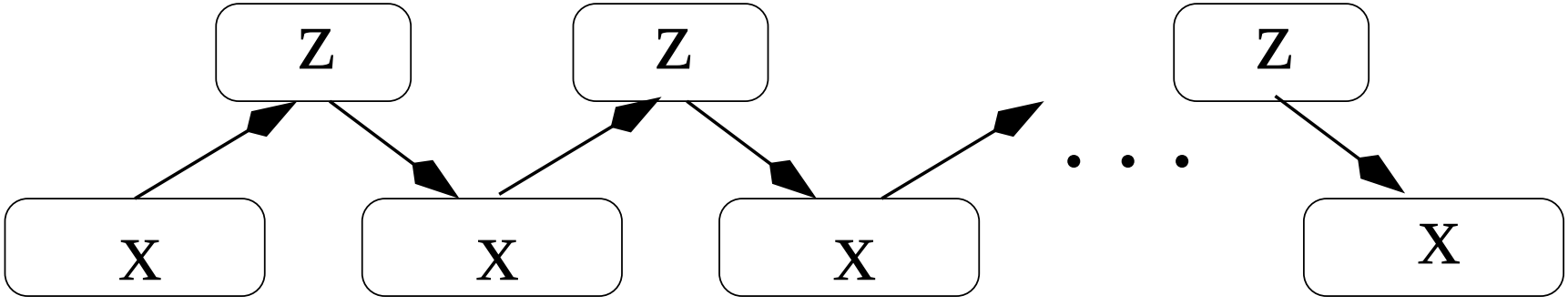
Deep Belief Networks

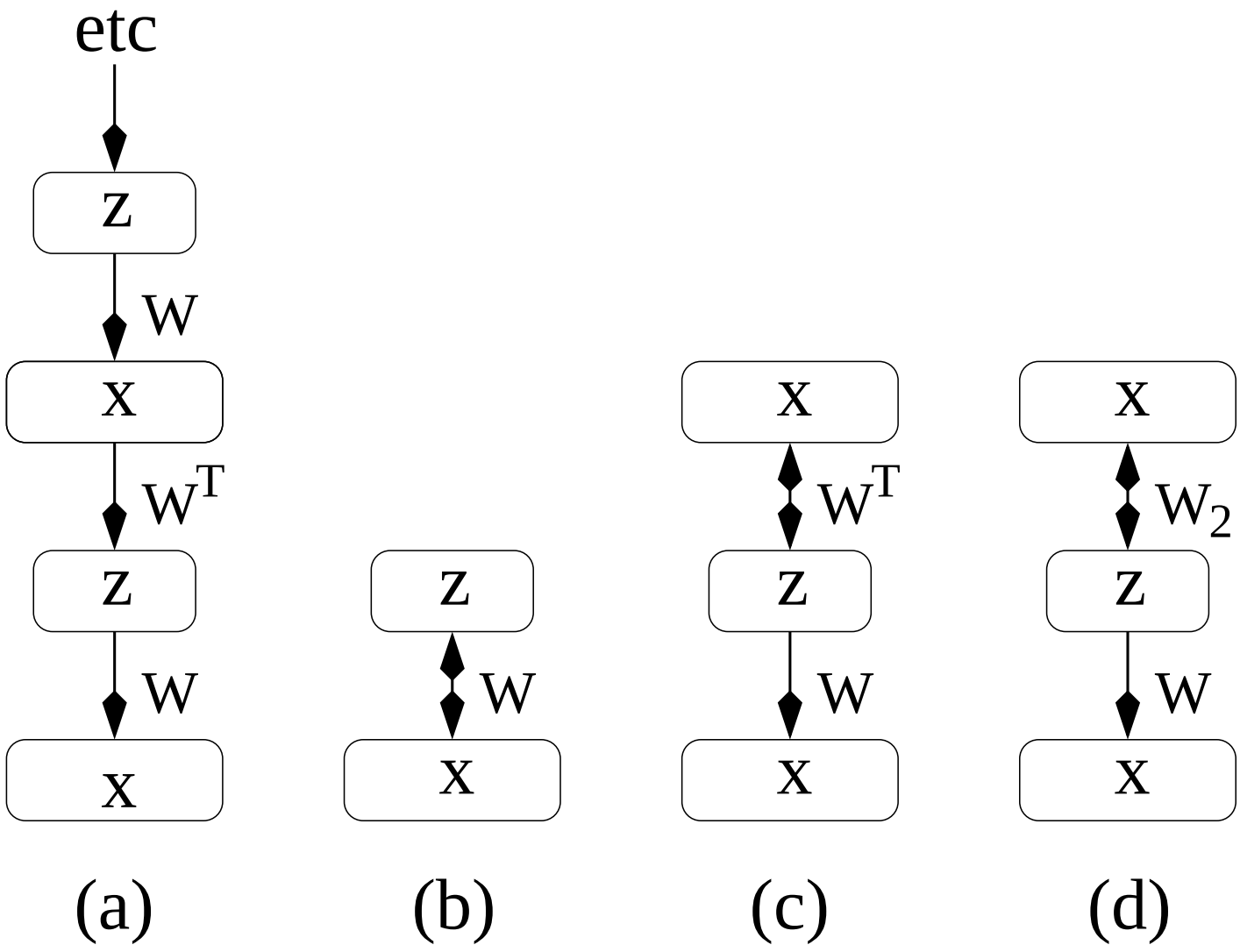
- Hinton (circa 2006) argues that greedy learning by stacking *directed* graphical models is problematic
- The parameters learned by the first layer model are such that the latent variables $\mathbf{z}$ should be independent
- But as this is not possible he argues this leads to a bad compromise, and that the aggregated posterior distribution $\langle p(\mathbf{z}|\mathbf{x})\rangle_{p(\mathbf{x})}$ may be no easier to model than $p(\mathbf{x})$
- He argues that using an *undirected* model (aka a Restricted Boltzmann Machine, RBM) as the basic learning module should be more effective
- Inference in an RBM is easy; each hidden unit can be sampled independently given the input. Cf explaining away in directed models



undirected

To draw samples from this model we can use a Markov chain Monte Carlo methods based on block Gibbs sampling





- (a), (b) and (c) are equivalent
- (d) is more powerful as W_2 can better model the correlations in the first hidden layer

Example: DBN learning on natural image patches

Lee, Ekanadham and Ng (2008)

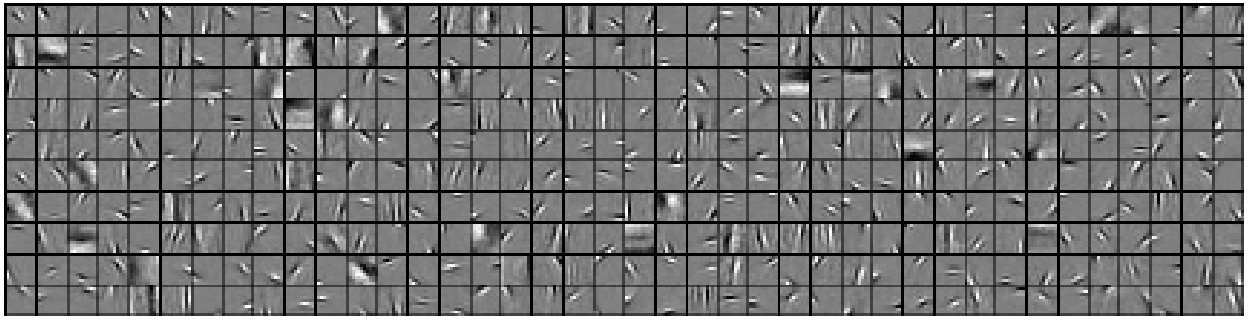
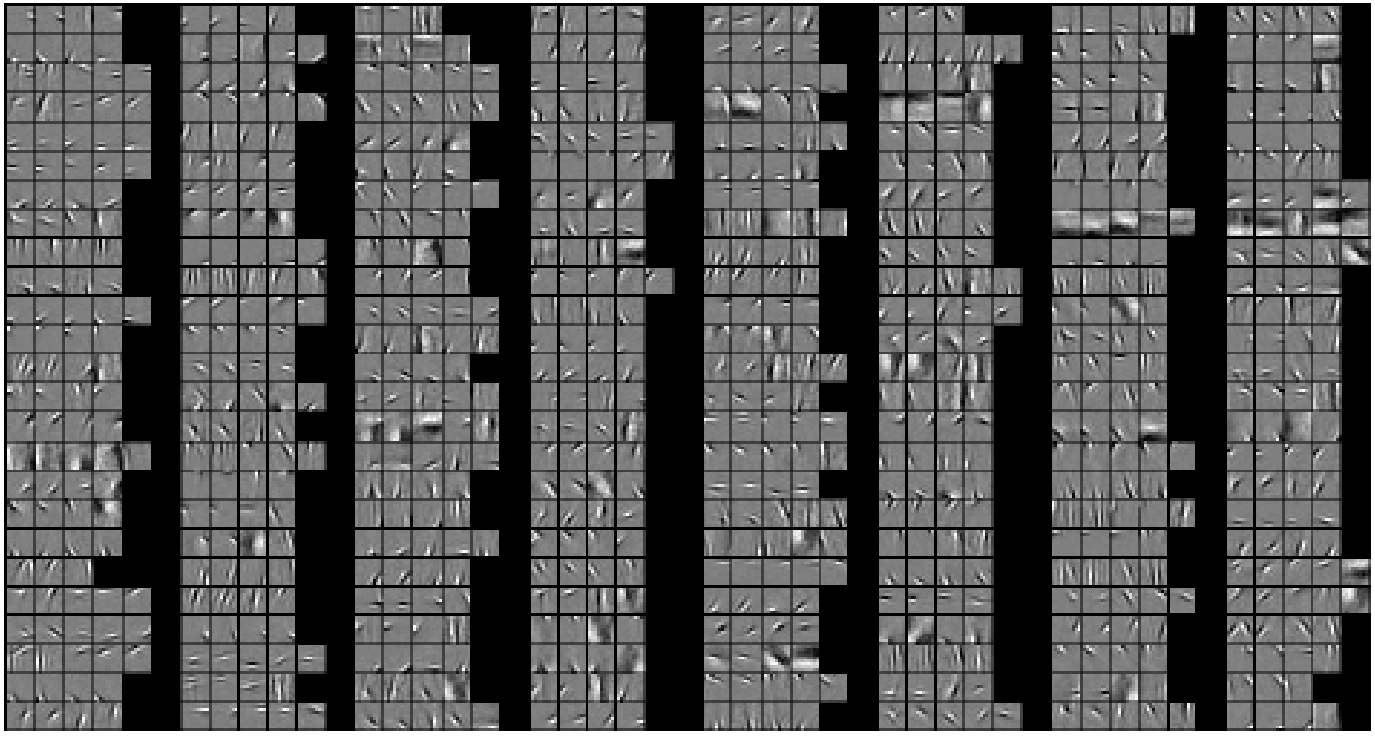
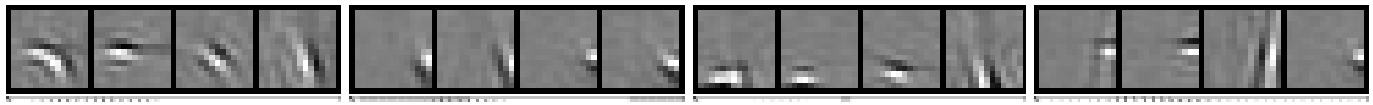


Figure 3: 400 first layer bases learned from the van Hateren natural image dataset, using our algorithm.





- A second layer unit responds to a combination of a few first layer units; see examples above
- Properties of “V2” units can be compared to neural data

- Comparison between methods can be based on e.g.
  - Visualization of learned weights
  - Evaluation of log likelihood under the model (technically difficult)
  - Use of the learned features for prediction tasks

Example: Acoustic modelling using DBNs

Mohammed, Dahl, Hinton (2010)

- In speech recognition, a Gaussian mixture model for mel frequency cepstral coefficients (MFCCs) is the standard acoustic modelling framework
- Mohammed et al show that they can obtain better performance on the TIMIT corpus by training a multi-layer DBN, and then translating this into a feedforward classifier network.
- This system outperforms previous methods on the TIMIT corpus

5. Discussion

Summary

- One-hidden-layer models have proved useful
- They can be readily extended through time
- Factors can interact in complex ways to create observations, e.g. via masks and occlusion
- Data generators can have rich hierarchical structure
- Greedy learning and DBNs are one way to attack this

Issues

- Such generic strategies might be insufficient; goal is to discover representations of the data that compactly describe regularities in it.
- Handling invariances
- Forms of factor interaction (e.g. gating); higher-order units
- Relationship between the factors and the data may be complicated and non-linear, but not noisy
- Variable architecture to provide explicit grouping/ownership of chunks of the data
- Technical challenges: inference, learning, model comparison
- Problems waiting to be solved!