Deep Gaussian Processes

Learning Abstract Features with Gaussian Process Models

Neil D. Lawrence
Departments of Neuro- and Computer Science, University of Sheffield, U.K.

Max Planck Institute, Tuebingen

11th March 2013

Outline

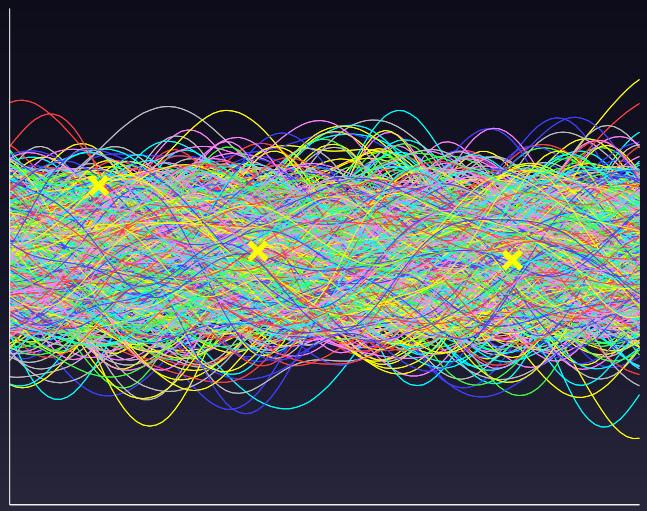
Template Models

Linear Dimensionality Reduction

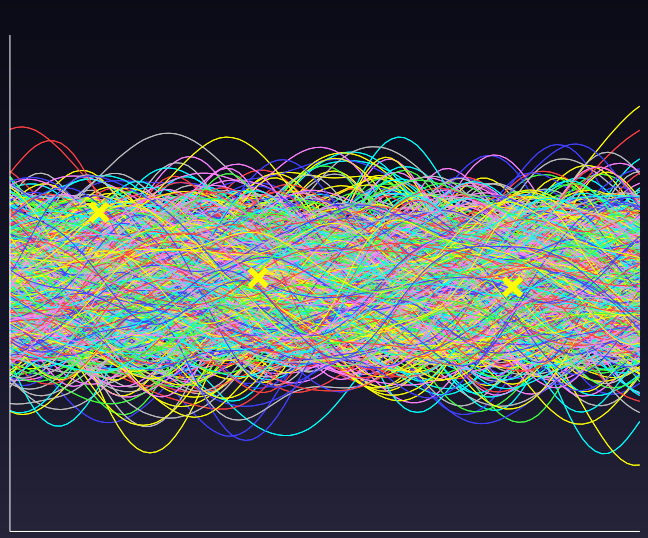
Deep Models

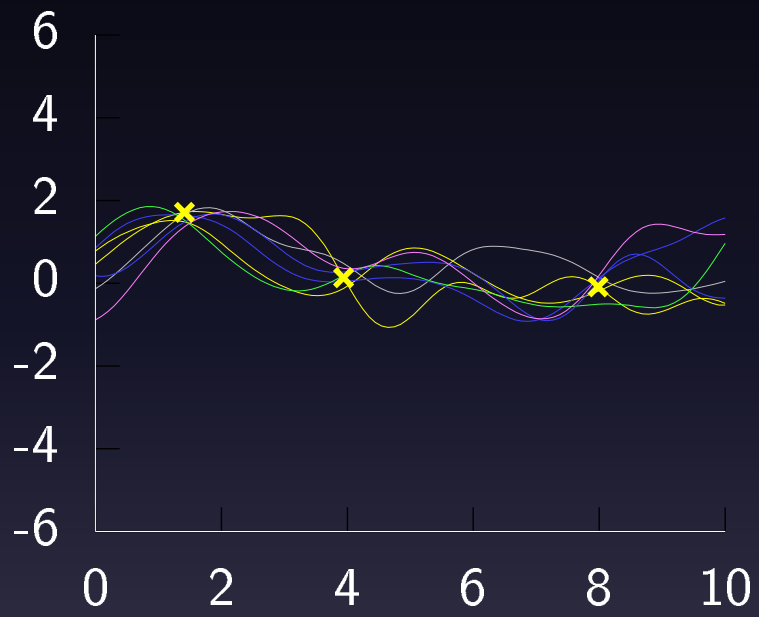
Conclusions

Gaussian Processes



Gaussian Processes





direction for further research.

11.1. HAVE WE THROWN THE BABY OUT WITH THE BATH WATER?

According to the hype of 1987, neural networks were meant to be intelligent models which discovered features and patterns in data. Gaussian processes in contrast are simply smoothing devices. How can Gaussian processes possibly replace neural networks? What is going on?

I think what the work of Williams and Rasmussen (1996) shows is that many real–world data modelling problems are perfectly well solved by sensible smoothing methods. The most interesting problems, the task of feature discovery for example, are not ones which Gaussian processes will solve. But maybe multilayer perceptrons can't solve them either. On the other hand, it may be that the limit of an infinite number of hidden units, to which Gaussian processes correspond, was a bad limit to take; maybe we should backtrack, or modify the prior on neural network parameters, so as to create new models more interesting than Gaussian processes. Evidence that this infinite limit has lost something compared with finite neural networks comes from the observation that in a finite neural network with more than one output, there are non–trivial correlations between the outputs (since they share inputs from common hidden units); but in the limit of an infinite number of hidden units, these correlations vanish. Radford Neal has suggested the use of non–Gaussian priors in networks with multiple hidden layers. Or perhaps a completely fresh start is needed, approaching the problem of machine learning from a paradigm different from the supervised feedforward mapping.

Structure of Priors

MacKay: NIPS Tutorial 1997 “Have we thrown out the baby with the bathwater?” (Published as MacKay, 1998) Also noted by (Wilson et al., 2012)

Deep Models

- Universal approximator arguments ignore interesting priors.
- Gaussian process priors are amazing, but still limited.
  - Struggle to learn unusual long range correlations
  - Makes covariance functions inappropriate for 'multitask learning'.

Motivation for Non-Linear Dimensionality Reduction

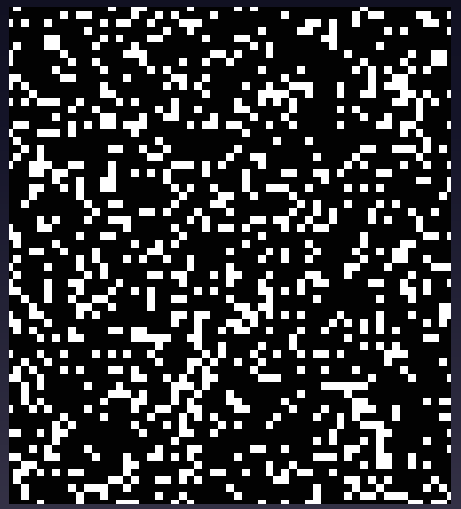
USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!

Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

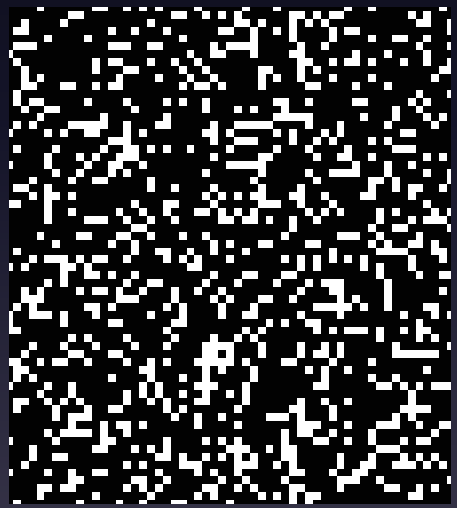
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

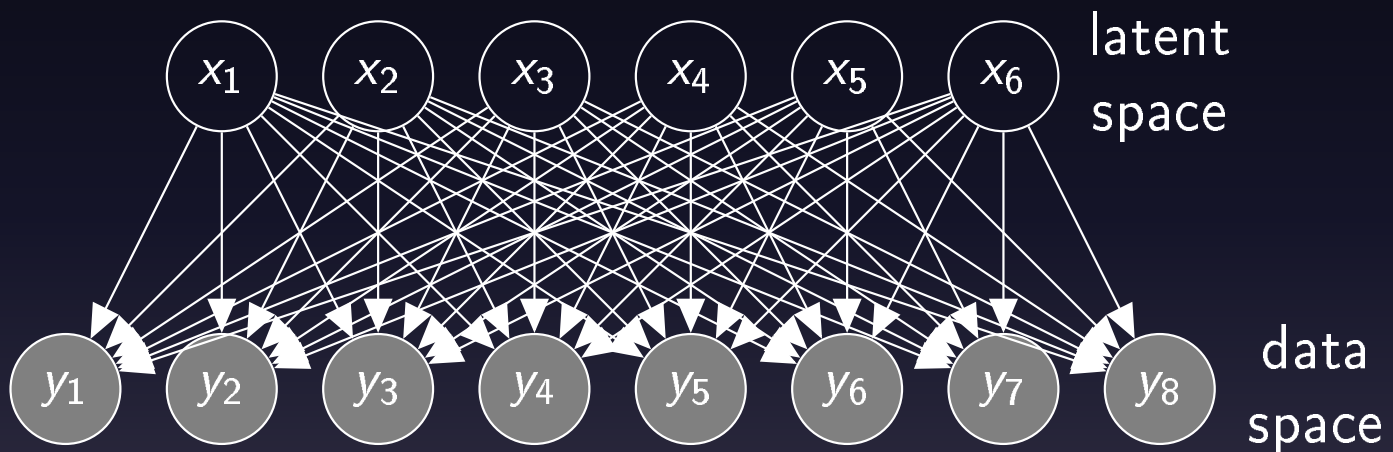
USPS Data Set Handwritten Digit

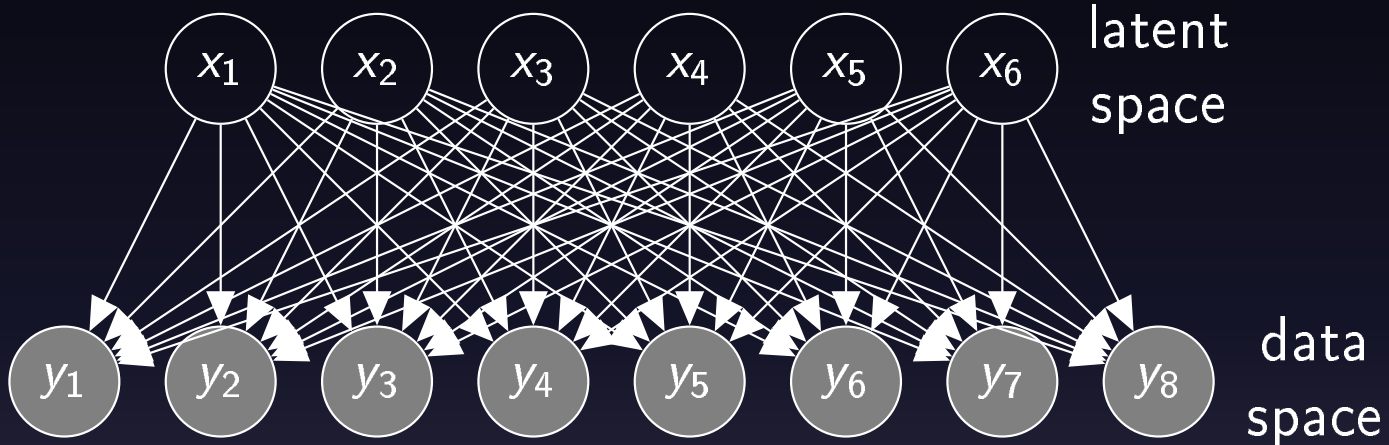
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!

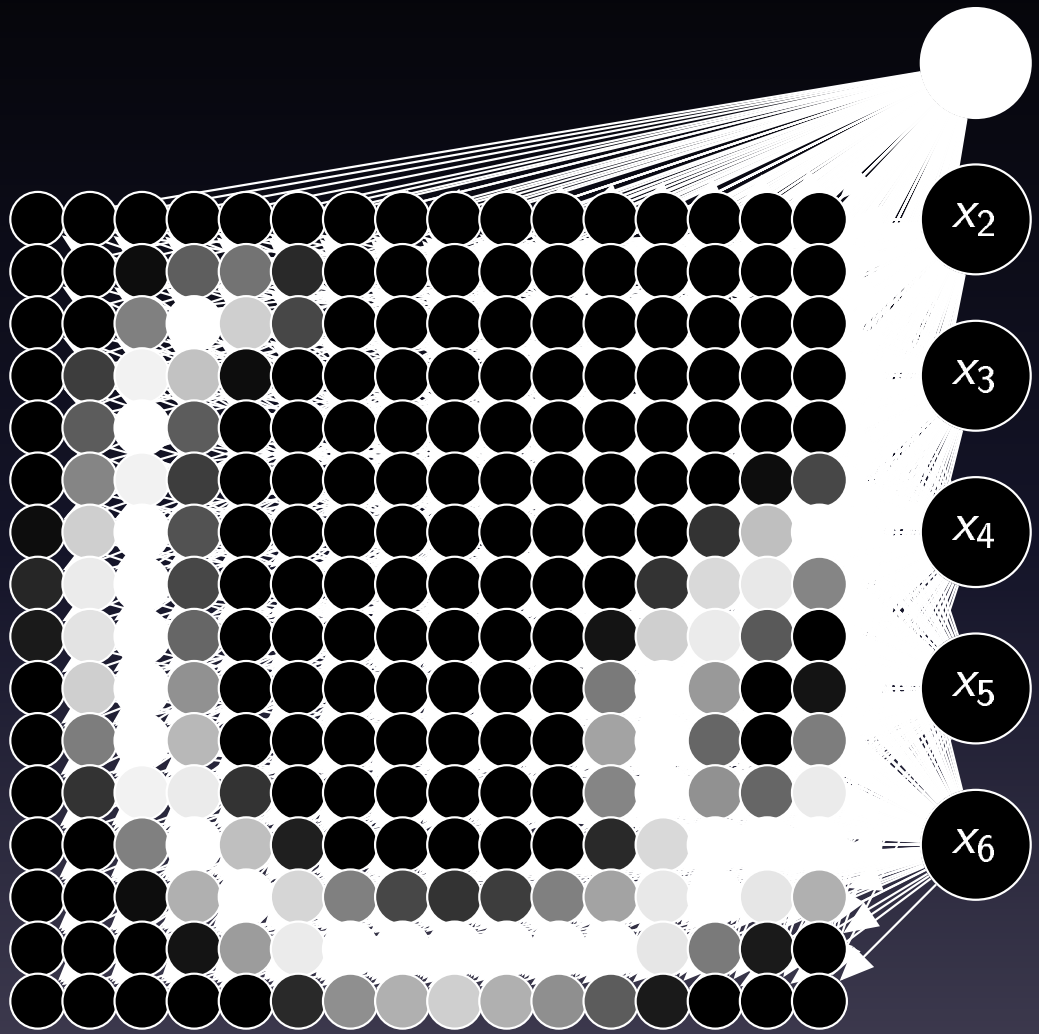
Template Model of Digits

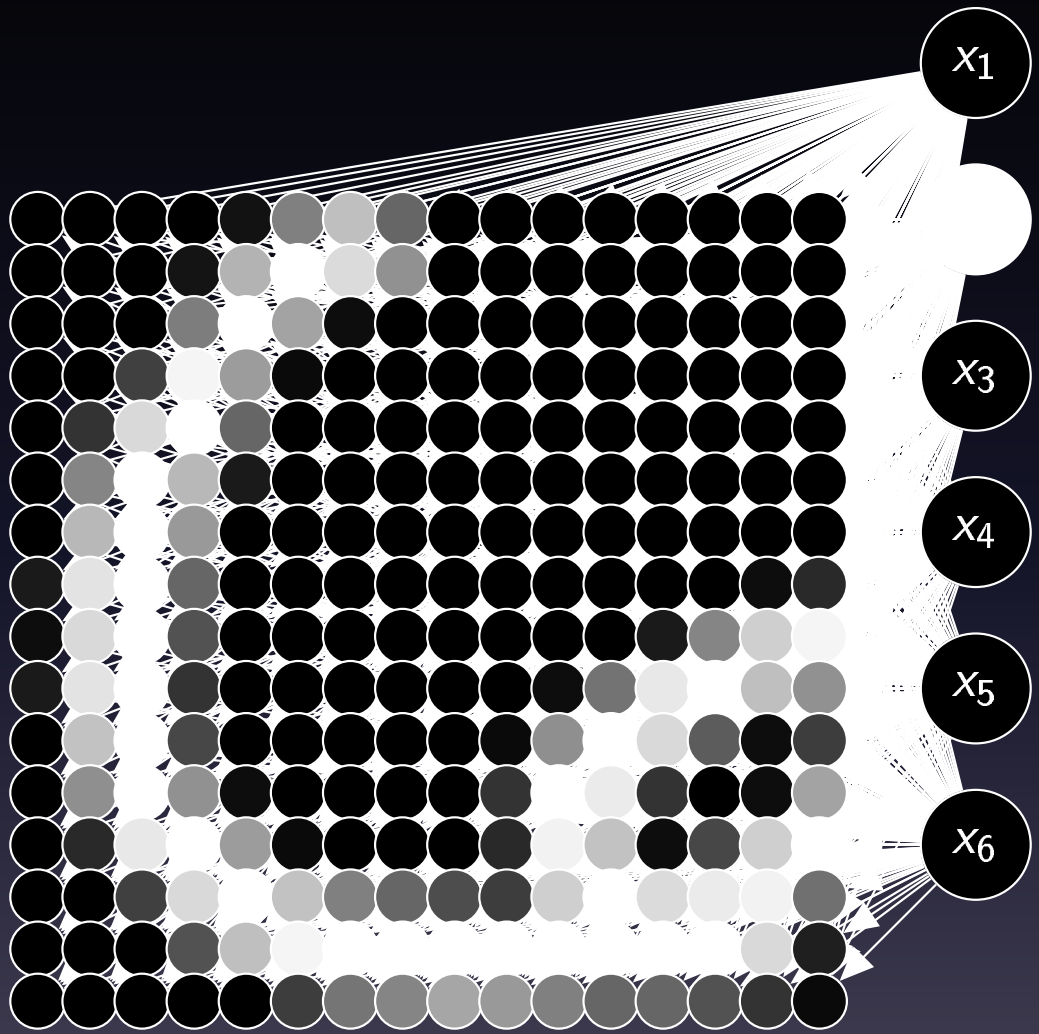
- Design a set of 'latent' features, which generate the 6.
- Global template: memorize data set.

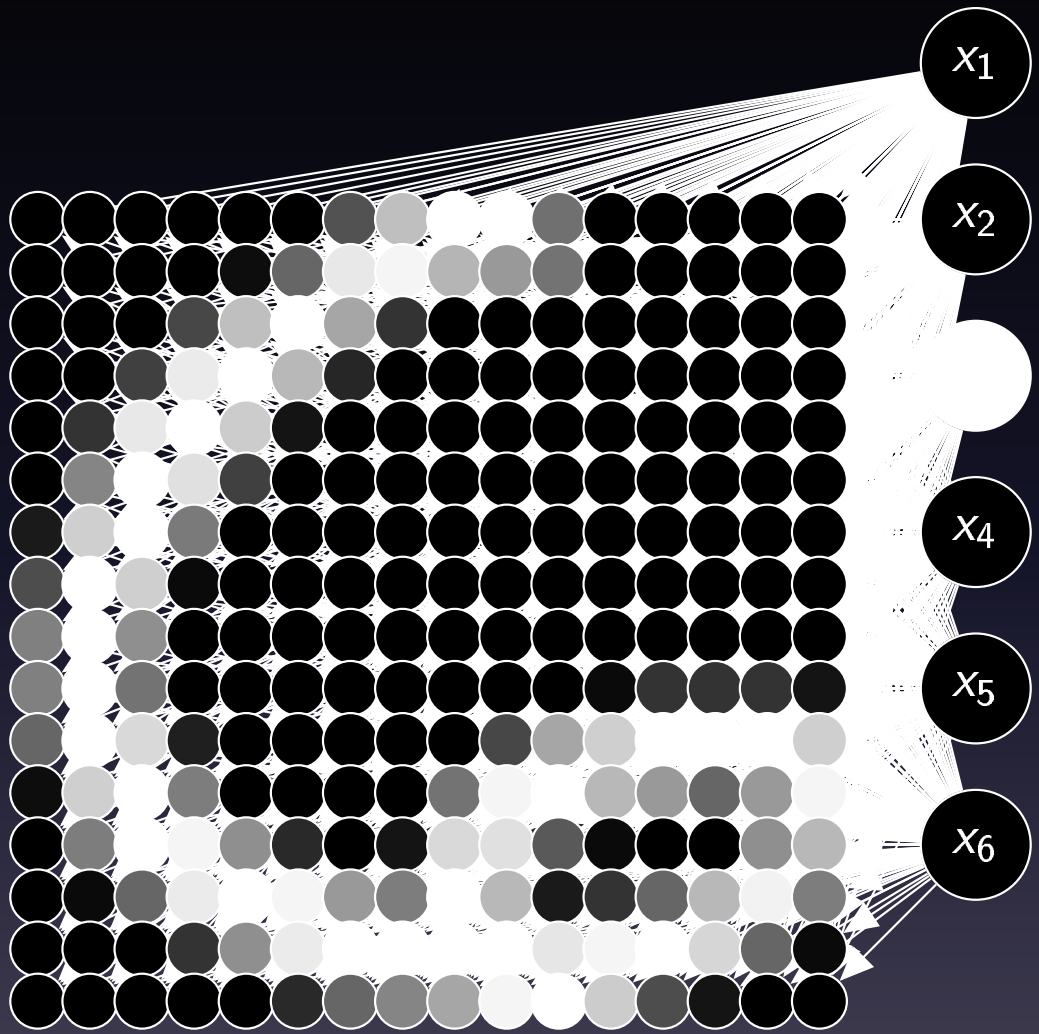
Latent Variable Model

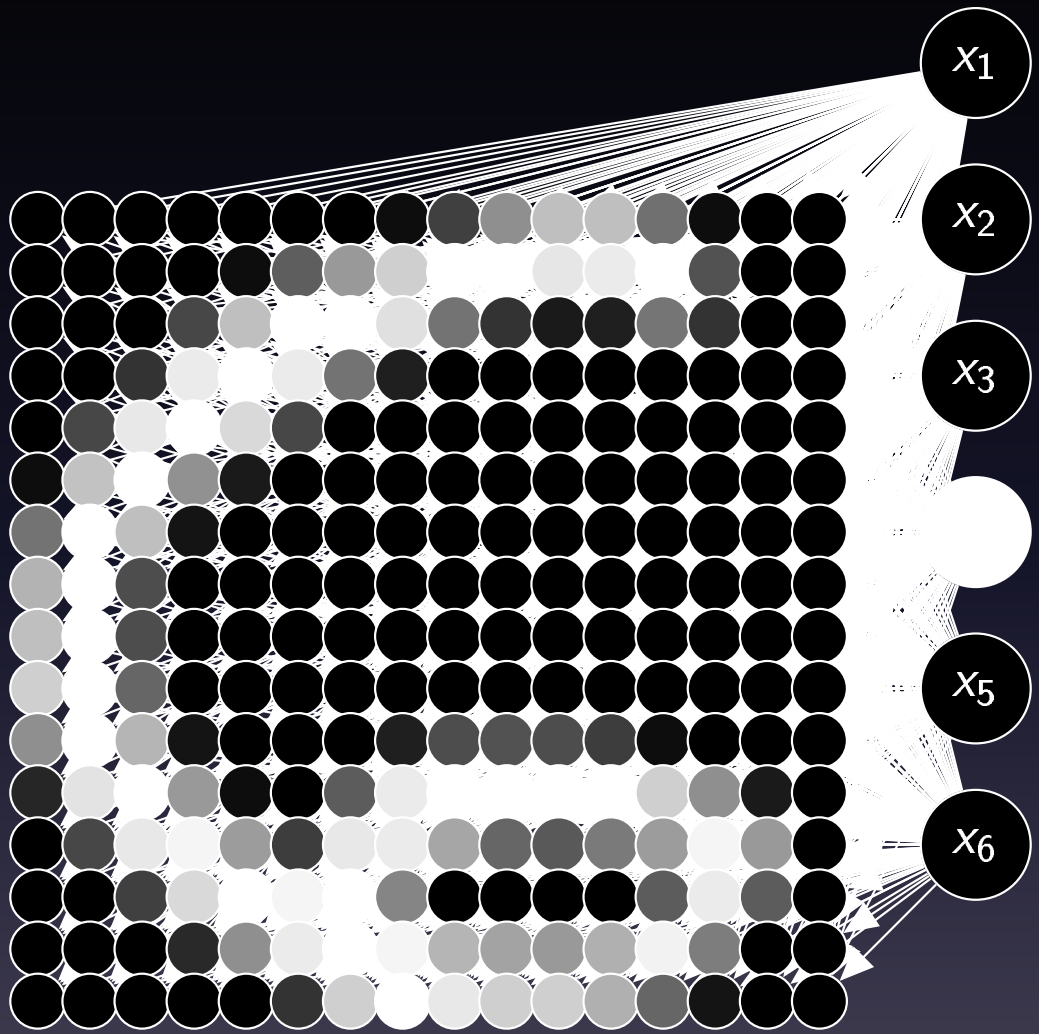


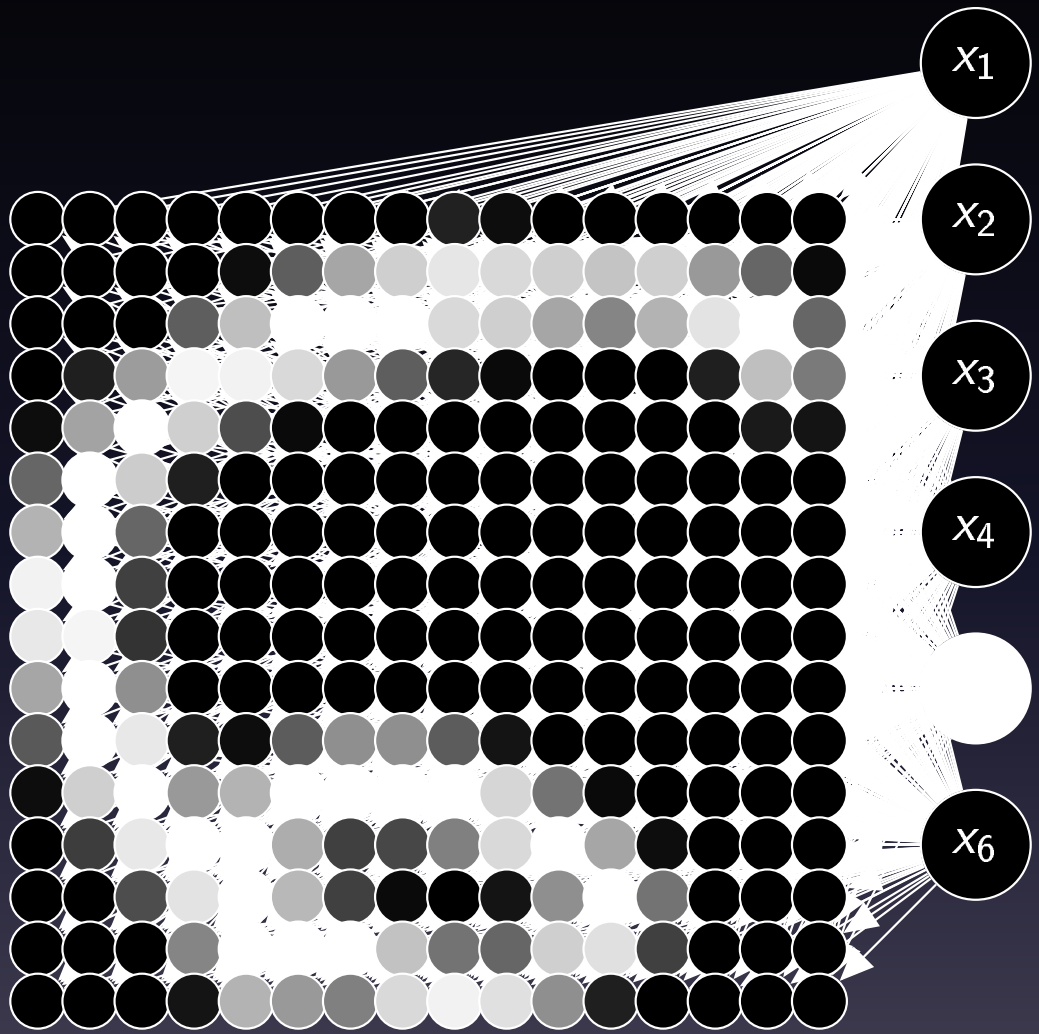

x_1
x_2
x_3
x_4
x_5
x_6
latent space
y_1
y_2
y_3
y_4
y_5
y_6
y_7
y_8
data space

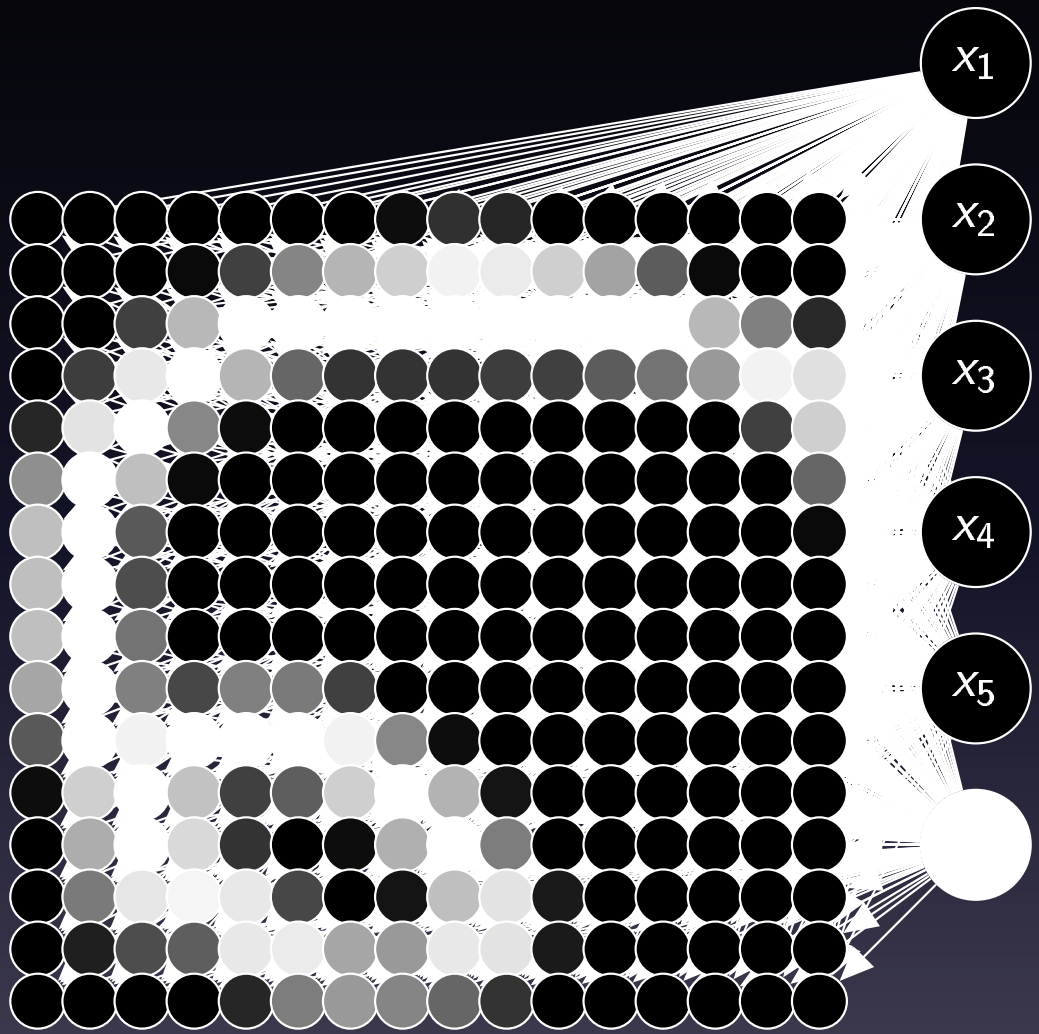

x_2
x_3
x_4
x_5
x_6


x_1
x_3
x_4
x_5
x_6


x_1
x_2
x_4
x_5
x_6


x_1
x_2
x_3
x_5
x_6


x_1
x_2
x_3
x_4
x_6


x_1
x_2
x_3
x_4
x_5

Template Matching

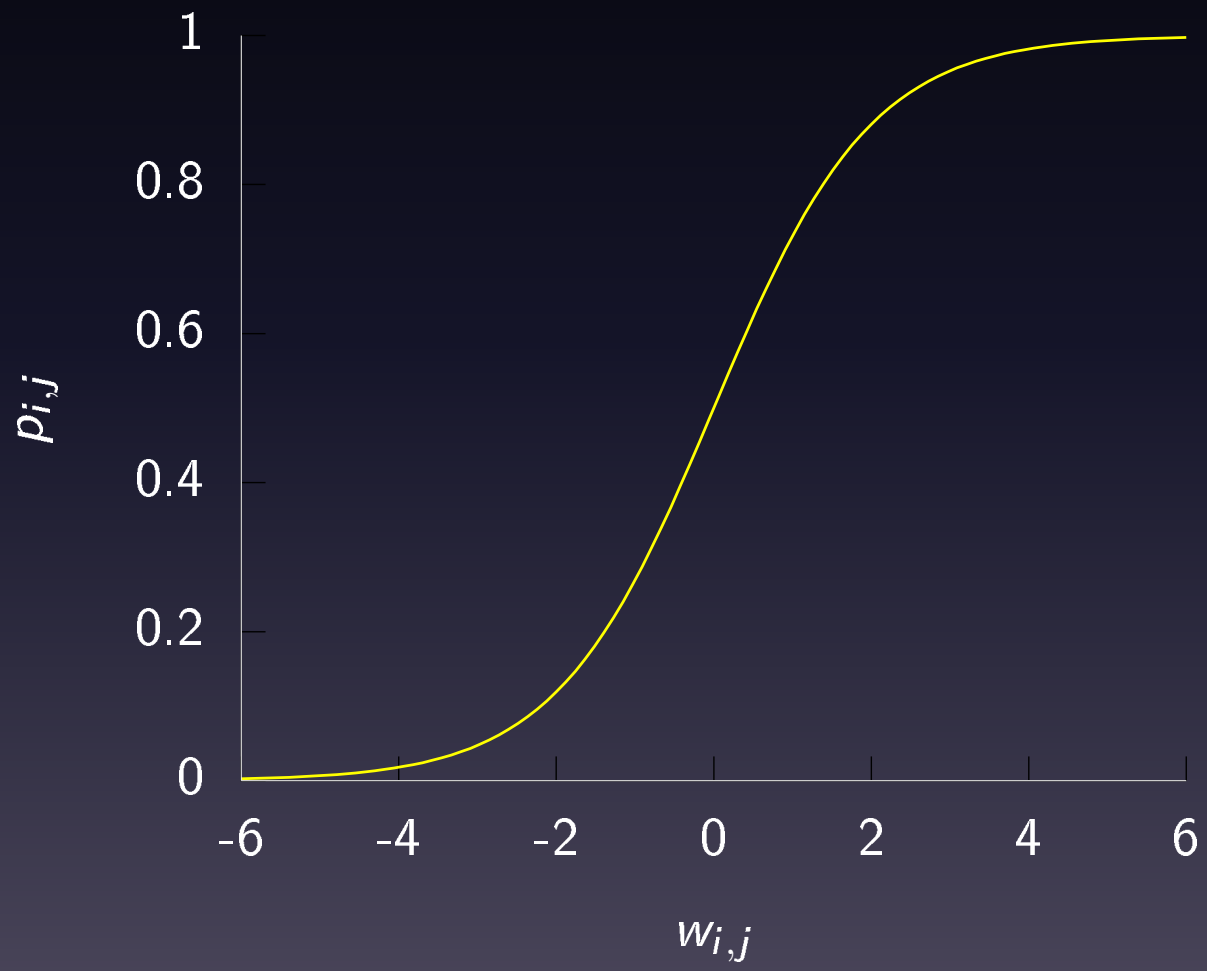
- Each latent node associated with a 'template' digit.
- If as many nodes as data then model is like 'nearest neighbour' with a particular distance measure.
- If less nodes than data then model is like a mixture of Bernoulli distributions.
- What if we allow several nodes to be switched on together?

Templates to Features

- In template matching ith node had an associated set of probabilities, $\mathbf{p}_i$.
- These probabilities can be reshaped into a matrix and sampled from to see the sixes.
- If the ith node is on the ith vector of probabilities is used.
- What if the ith node and the kth node are on?
    - How do we combine $\mathbf{p}_i$ and $\mathbf{p}_k$ to give probabilities of pixels?

Squashing Function

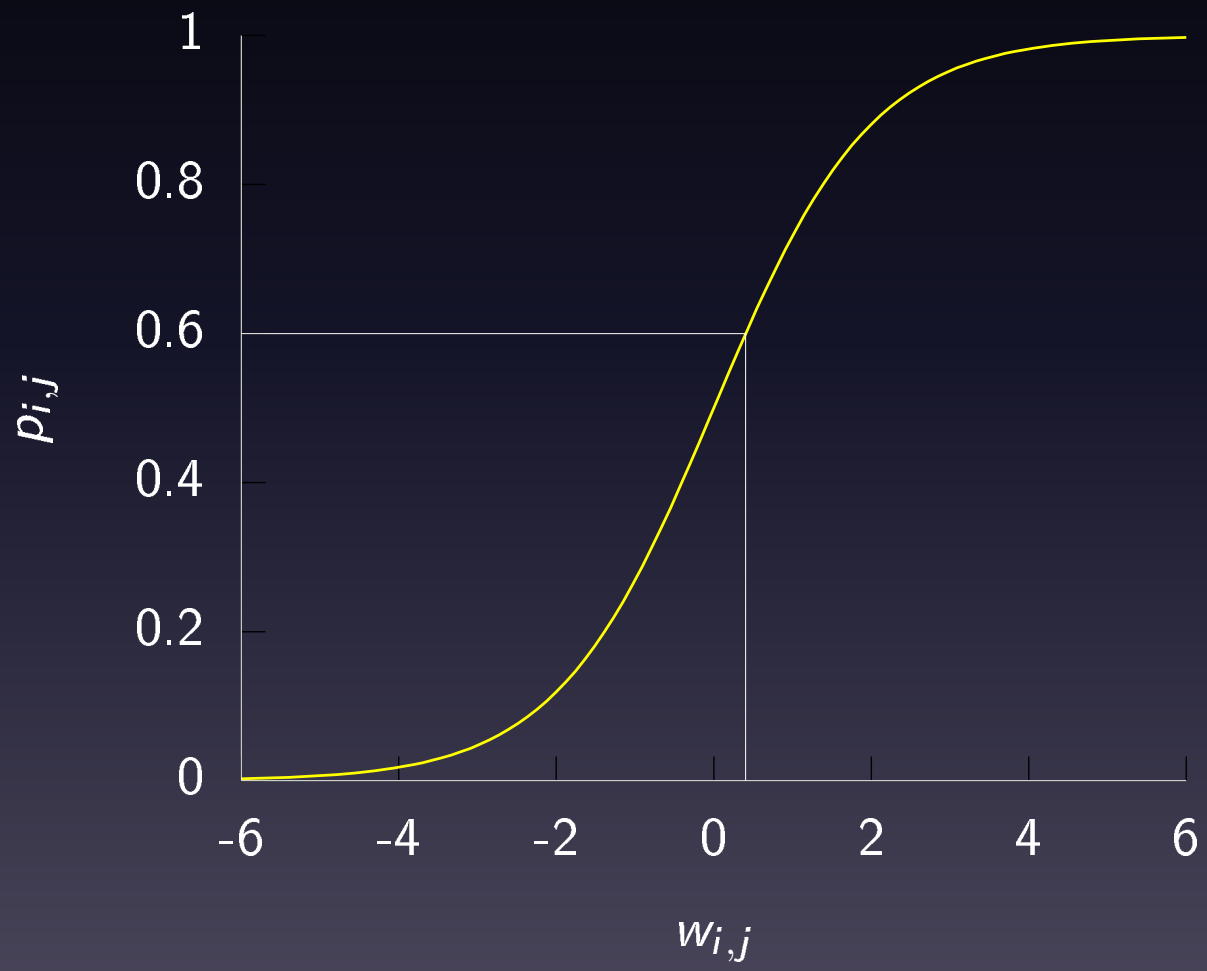
- One solution is to first reparameterise $p_{i,j}$ as a squashing function,



- For example the sigmoid function.

Squashing Function

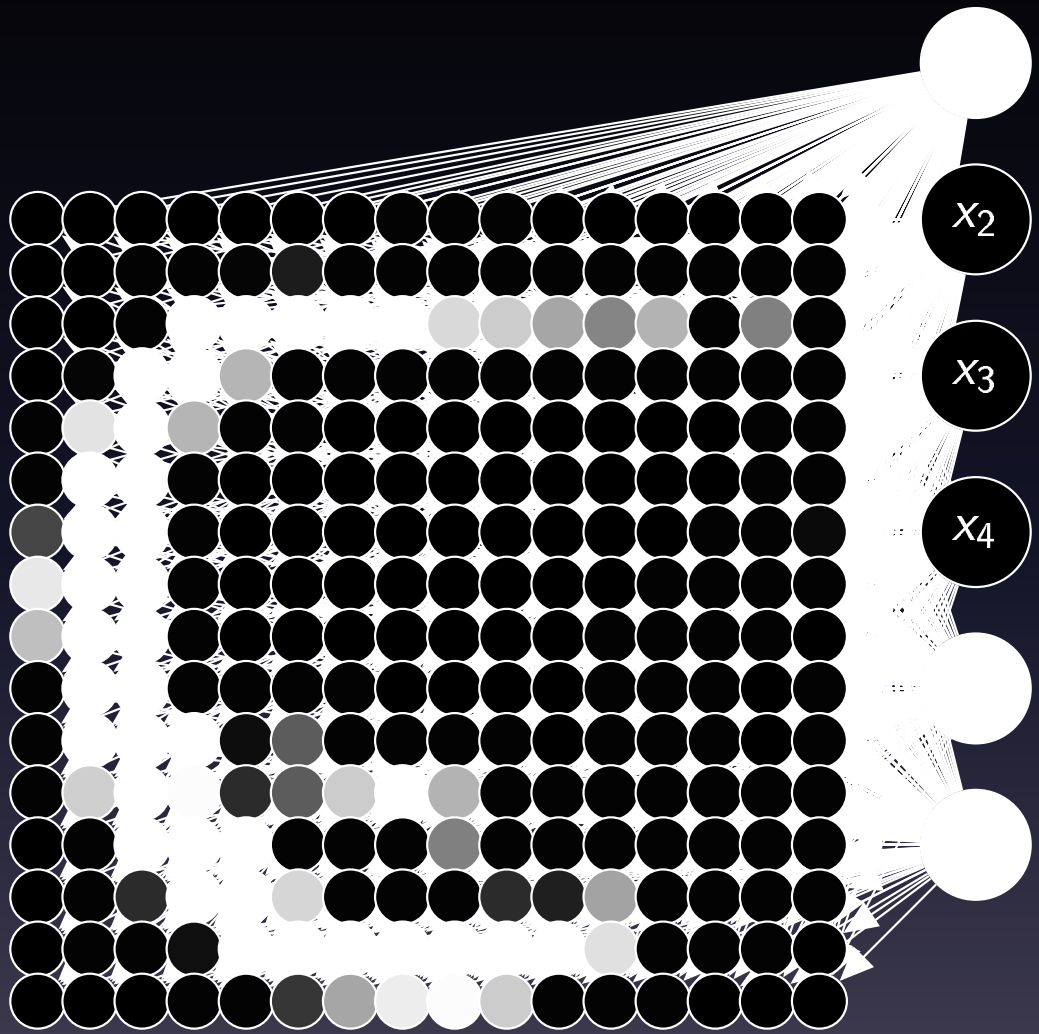
- One solution is to first reparameterise $p_{i,j}$ as a squashing function,

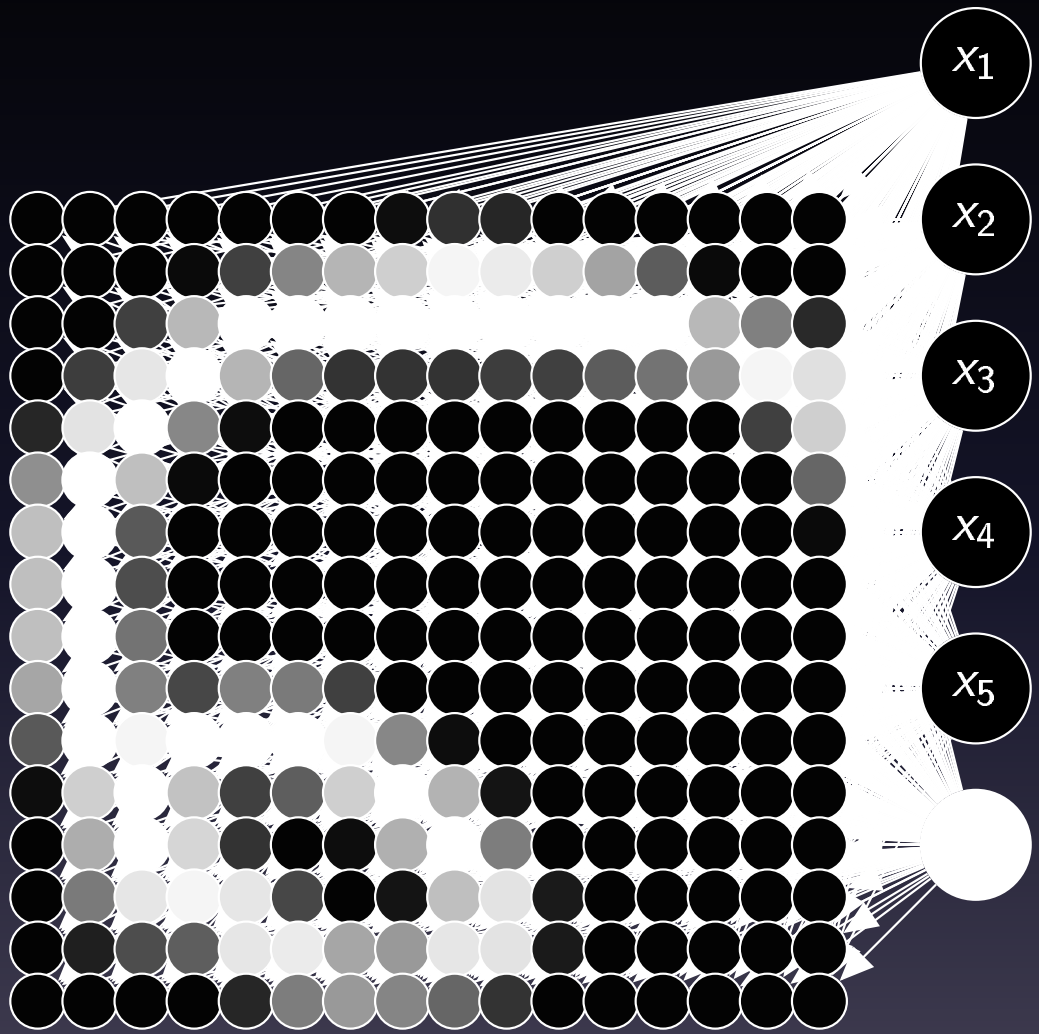


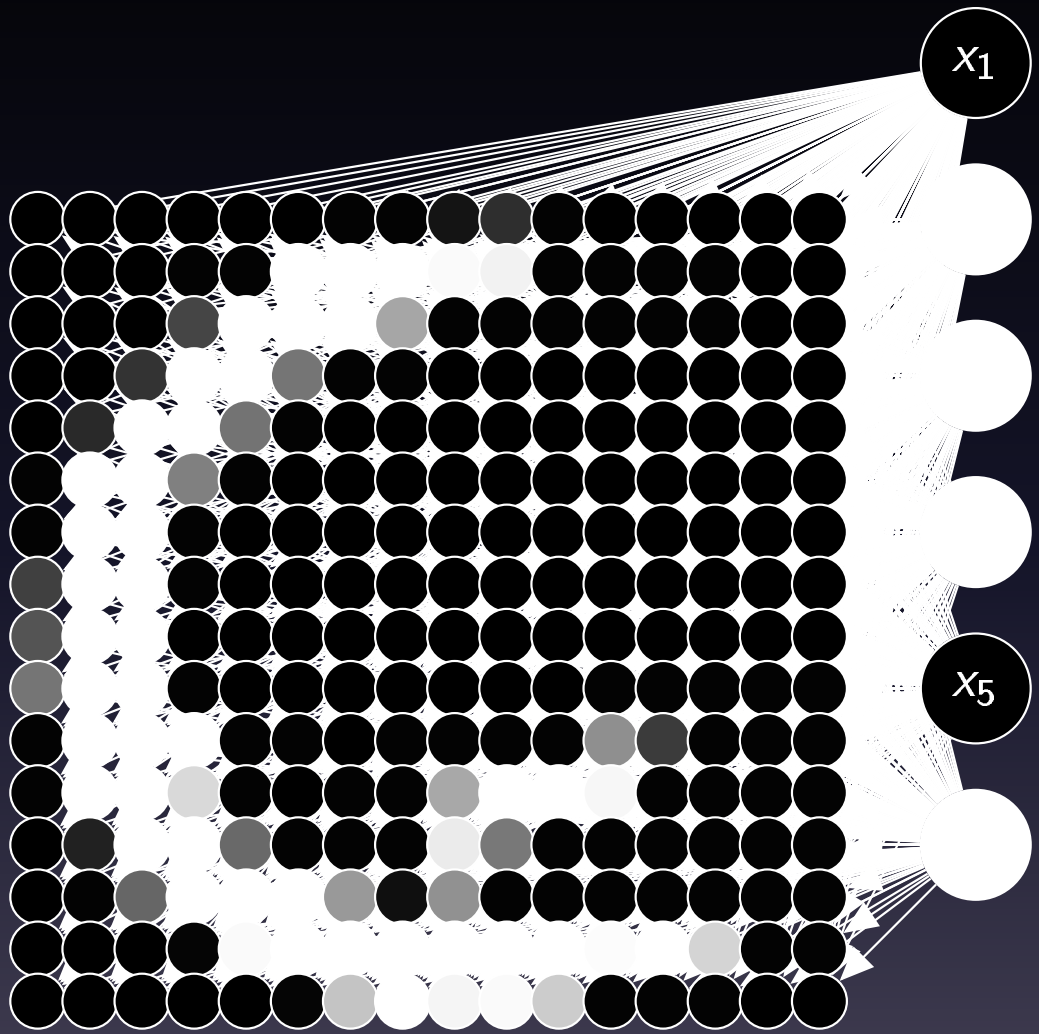
- For example the sigmoid function.

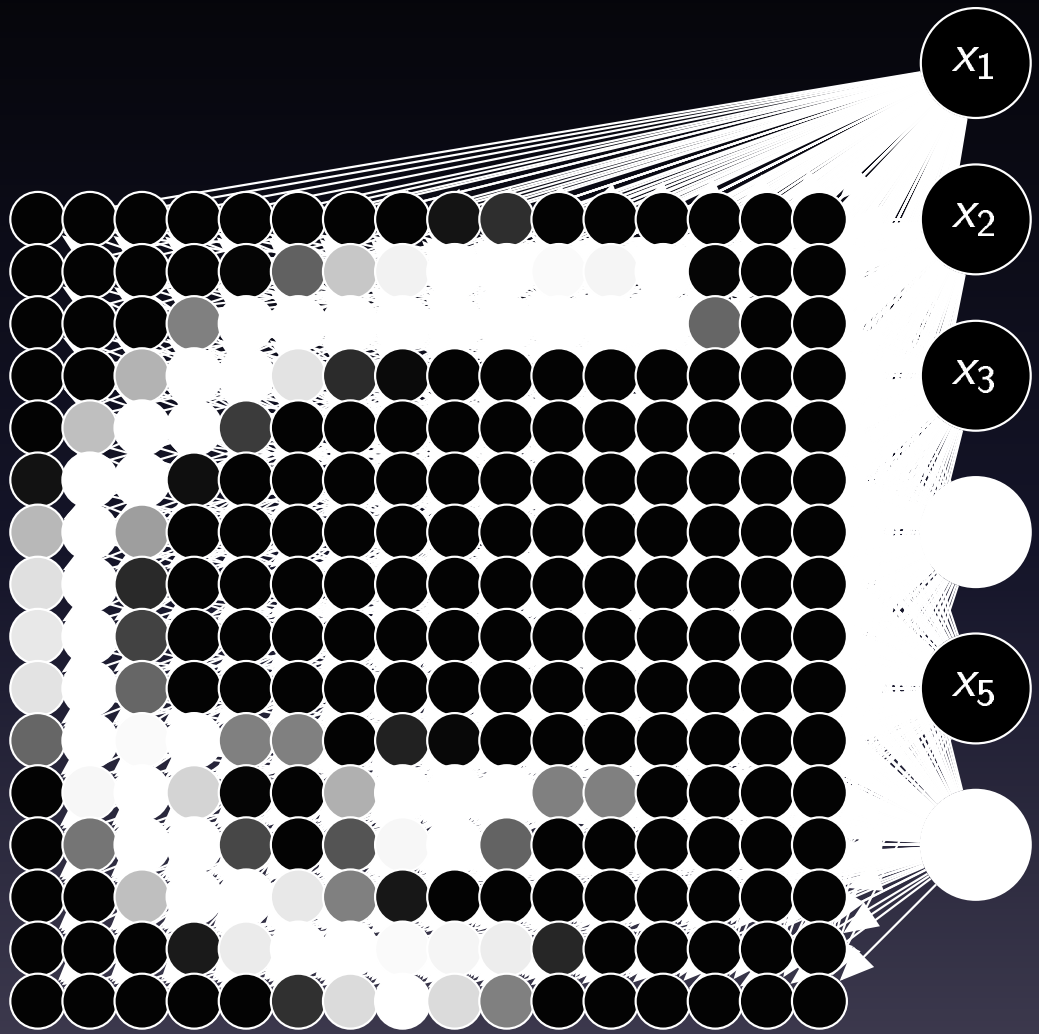
Addition Before Squashing

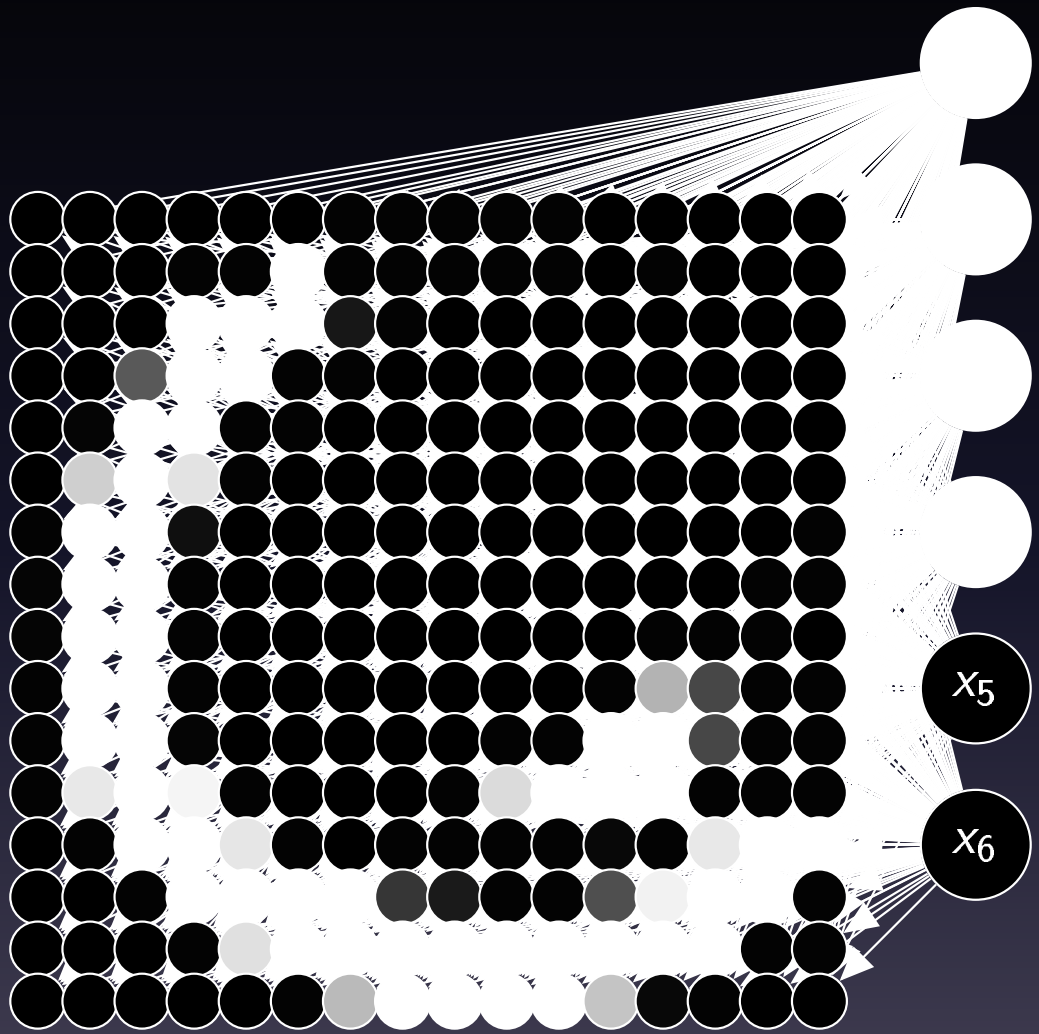
- Example: if latent node 1 and 6 are on.
- Can't add $p_{:,1}$ to $p_{:,6}$ to obtain probability that node is on.
- Instead add $w_{:,1}$ to $w_{:,6}$ and push through squashing function.
  - In general for $\mathbf{p}_{i,:}$ compute $\mathbf{W}\mathbf{x}_{i,:}$.
  - Then $p_{i,j} = \sigma(\mathbf{w}_{j,:}^\top \mathbf{x}_{i,j})$ where $\sigma(\cdot)$ is the sigmoid function.

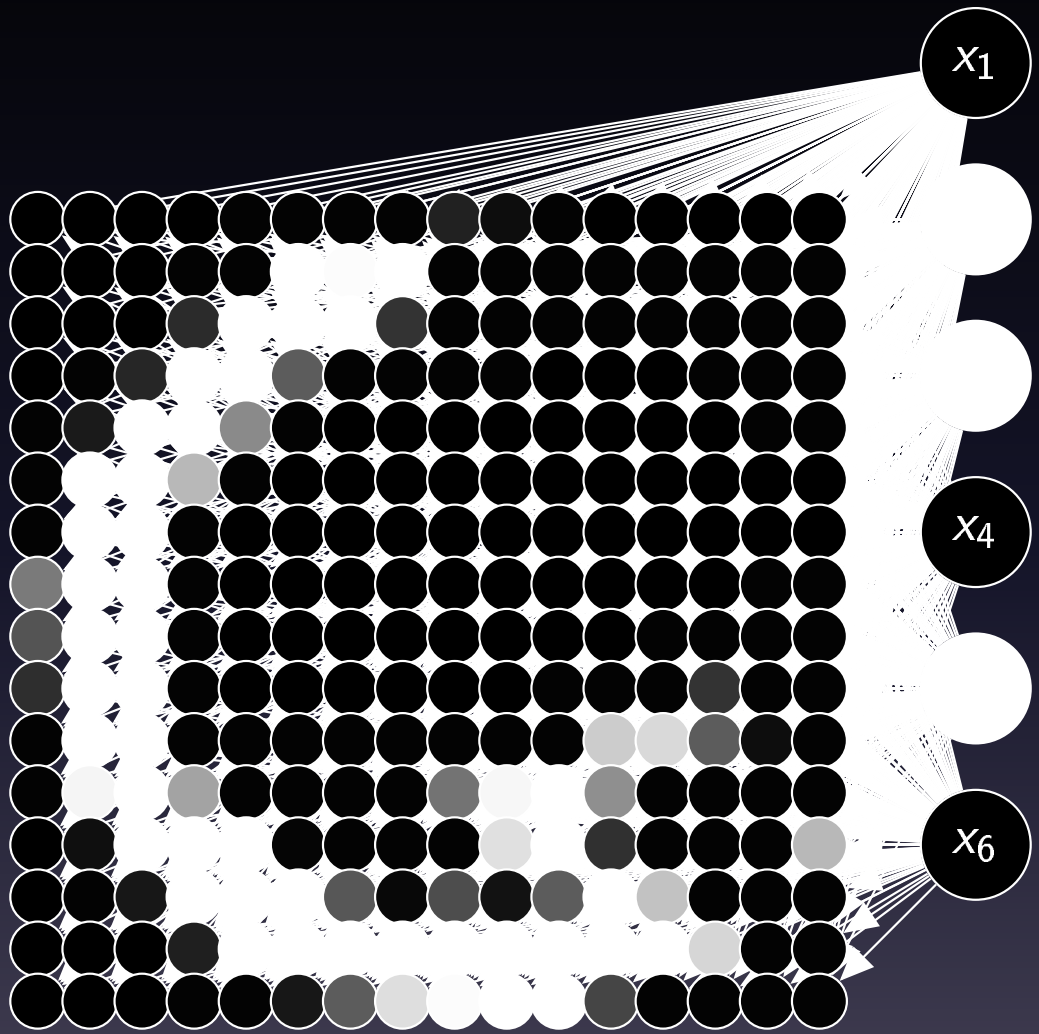

x_2
x_3
x_4


x_1
x_2
x_3
x_4
x_5


x_1
x_5

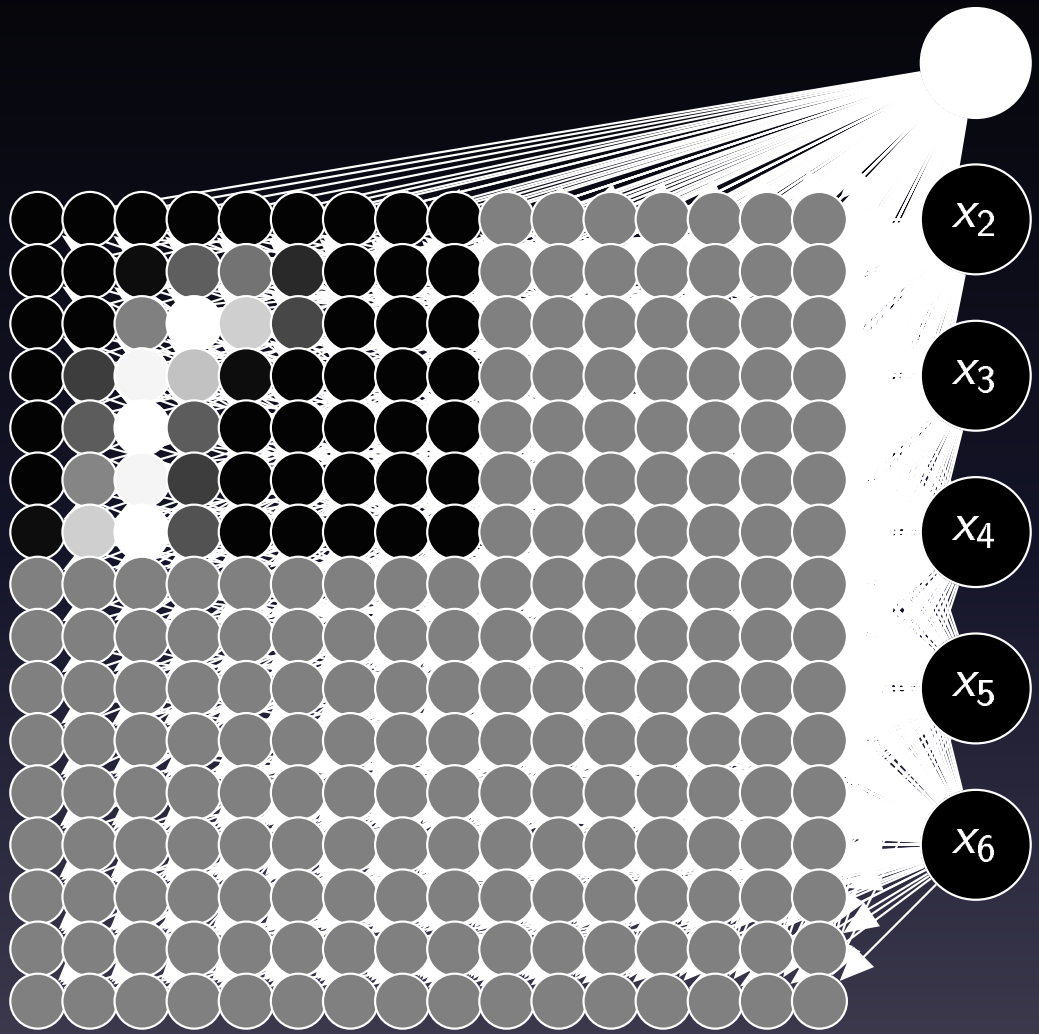

x_1
x_2
x_3
x_5

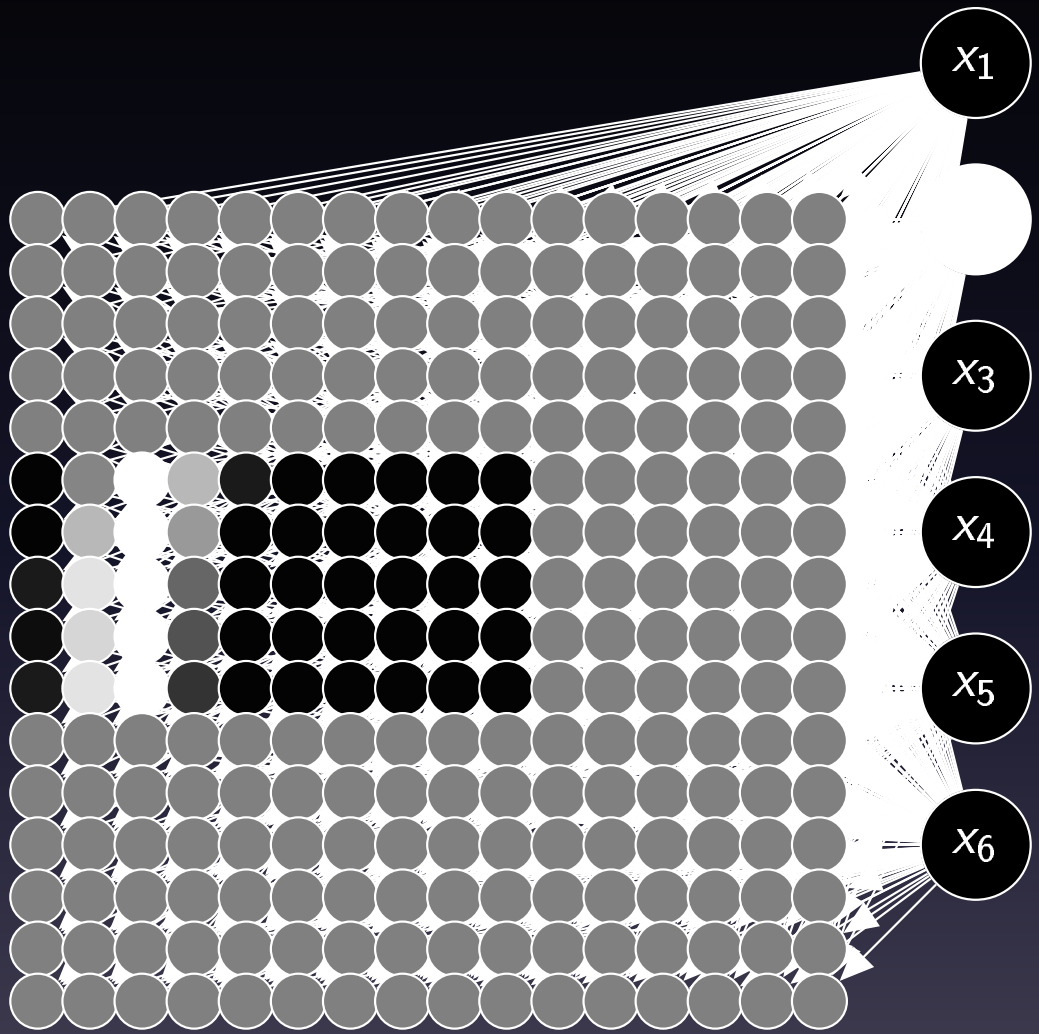

x_5
x_6

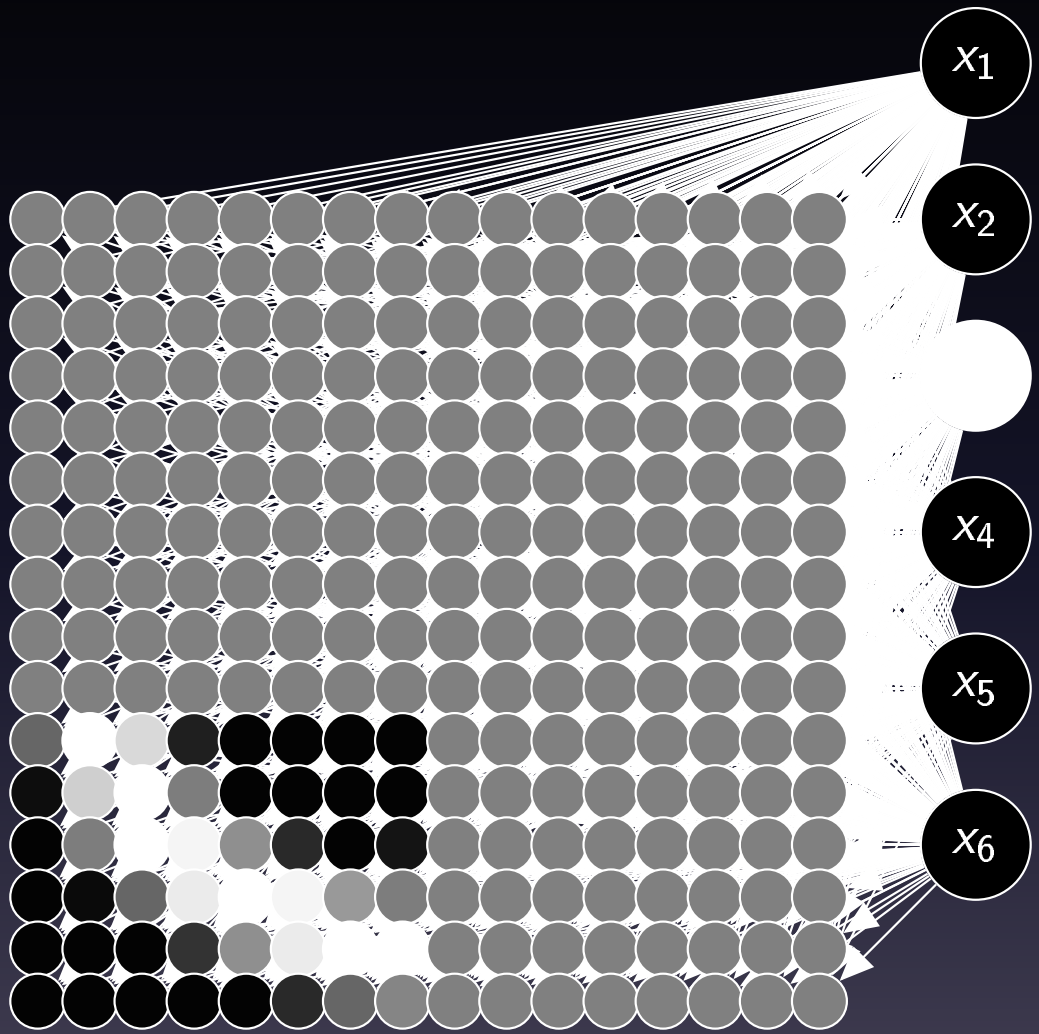

x_1
x_4
x_6

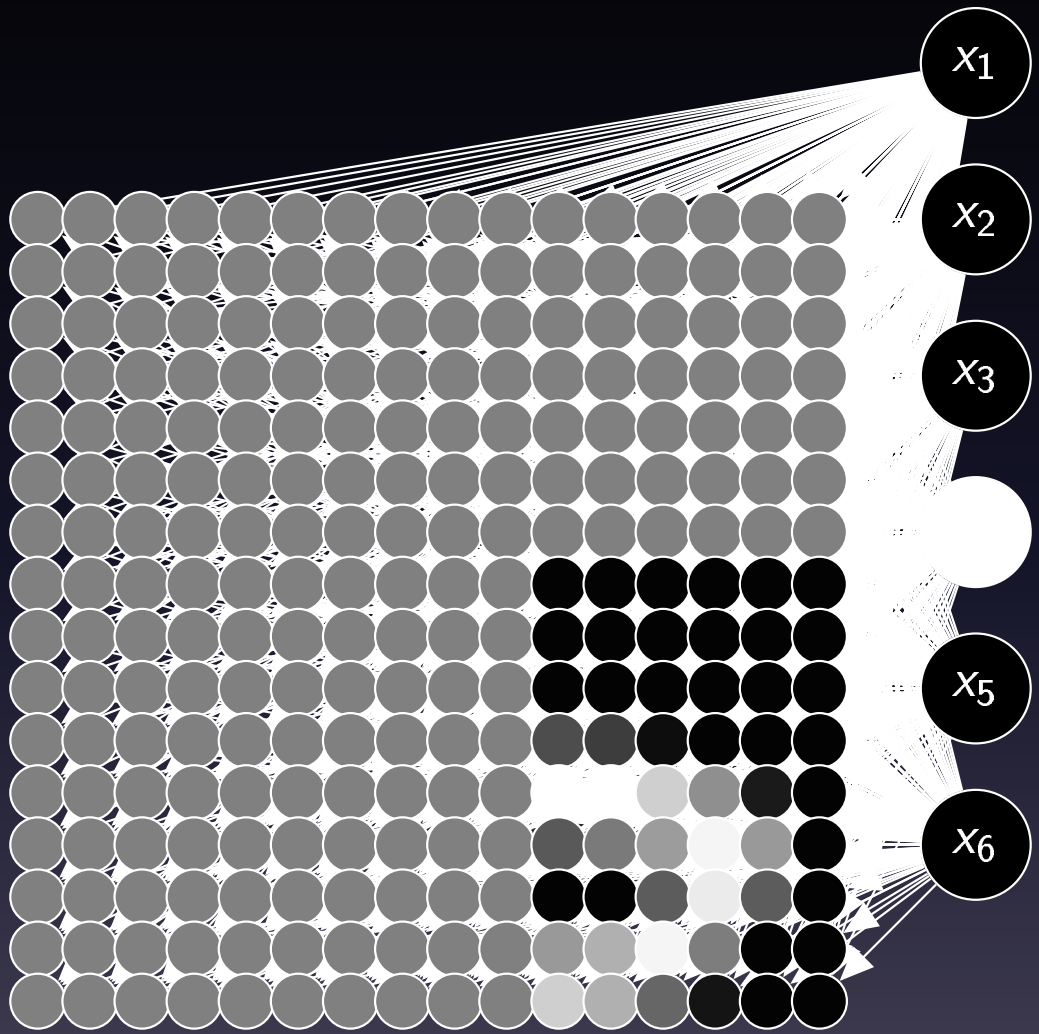
Localized Receptive Fields

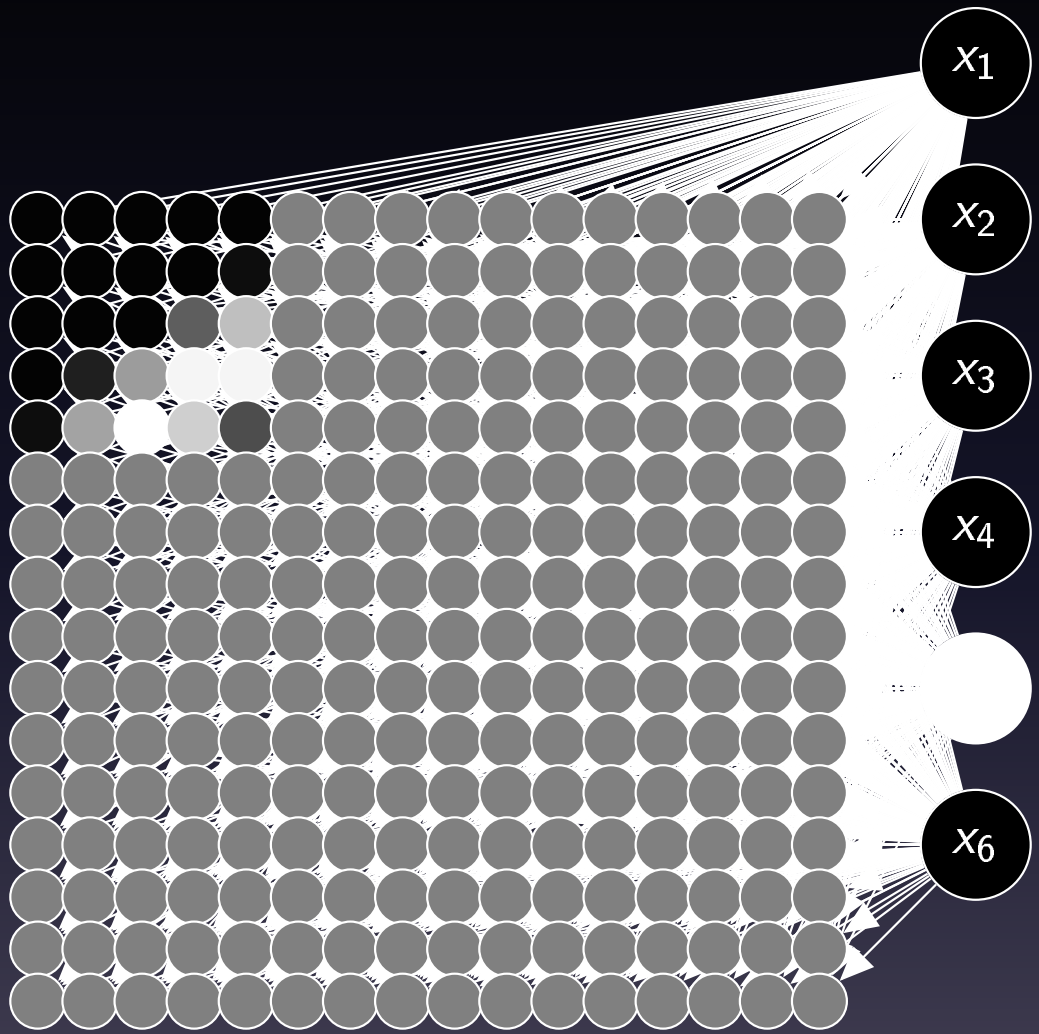
- Model can now fit global model as sum of parts.
- Each latent node associated with local features.
- Structure of model combines local features in products of experts manner (?).

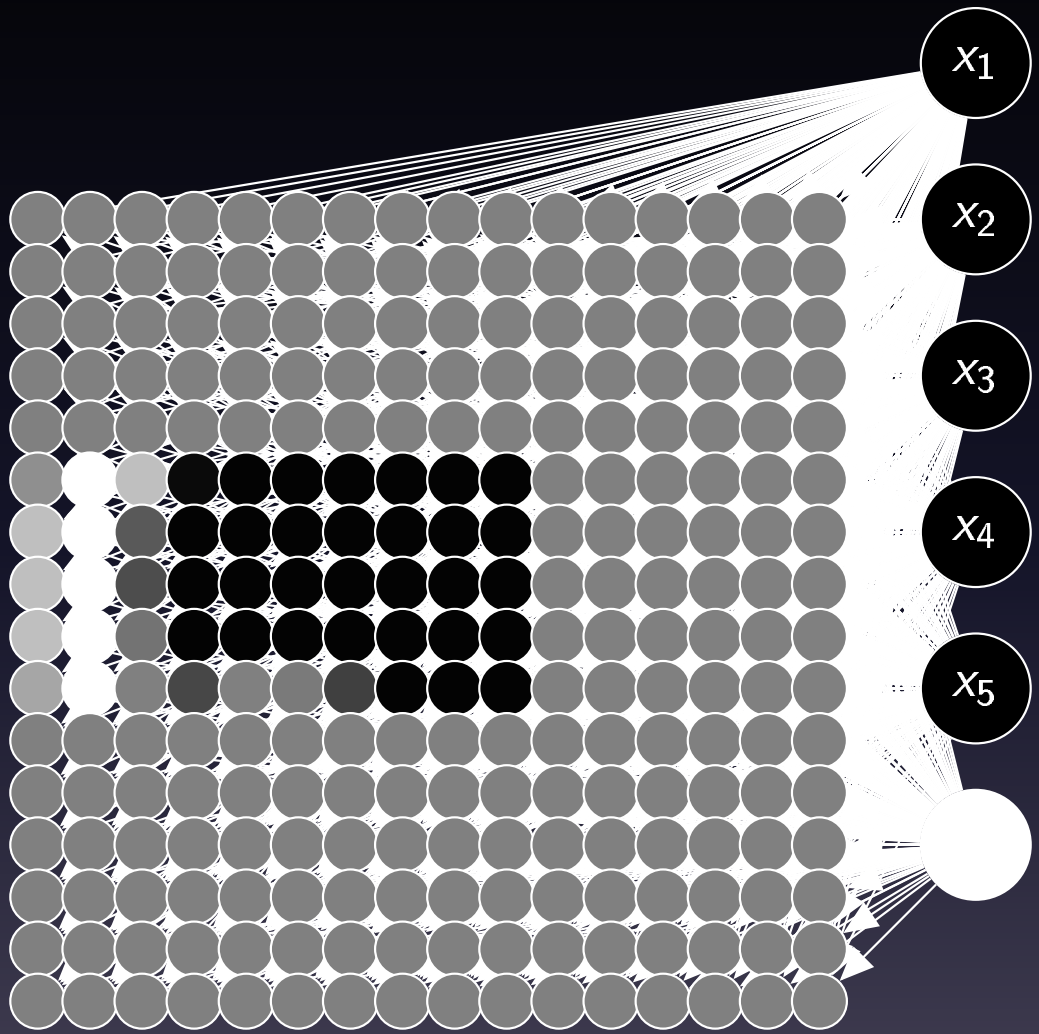

x_2
x_3
x_4
x_5
x_6

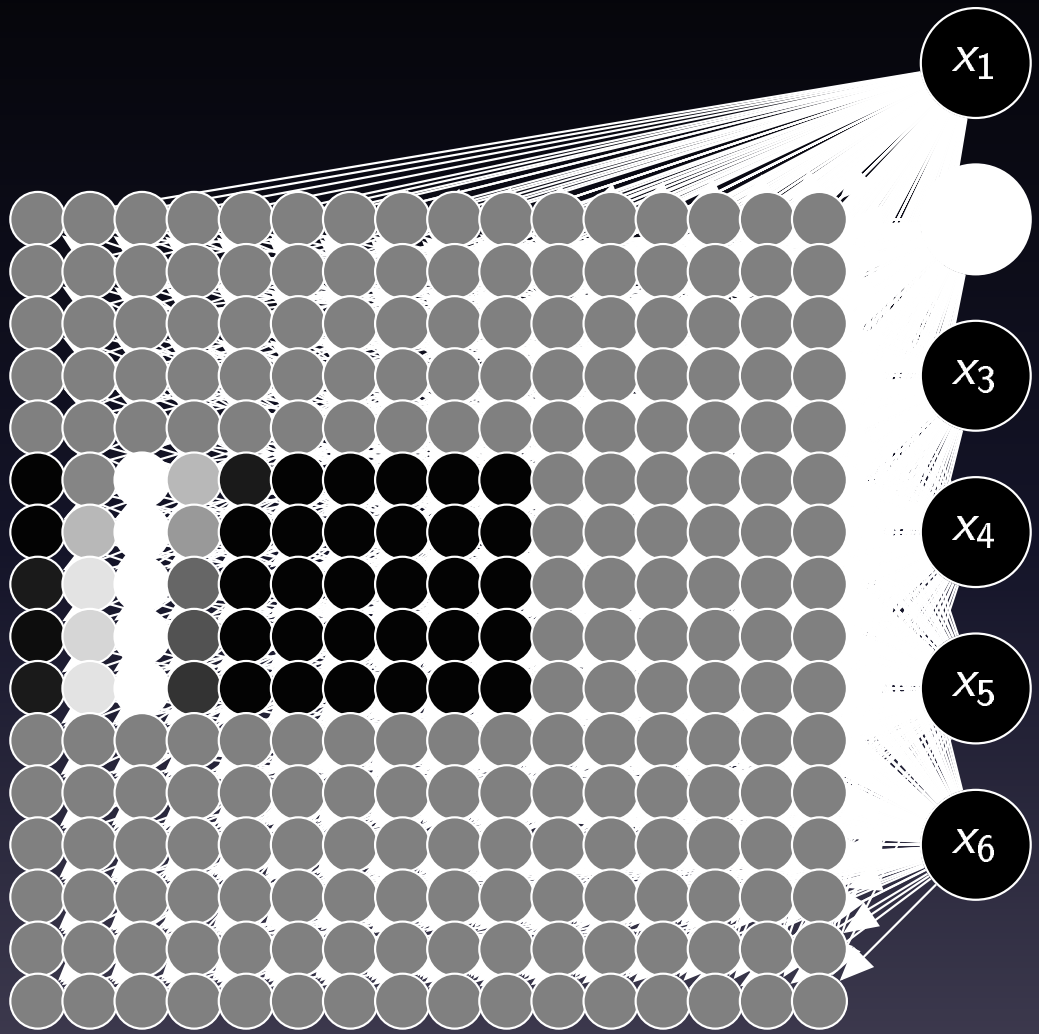

x_1
x_3
x_4
x_5
x_6

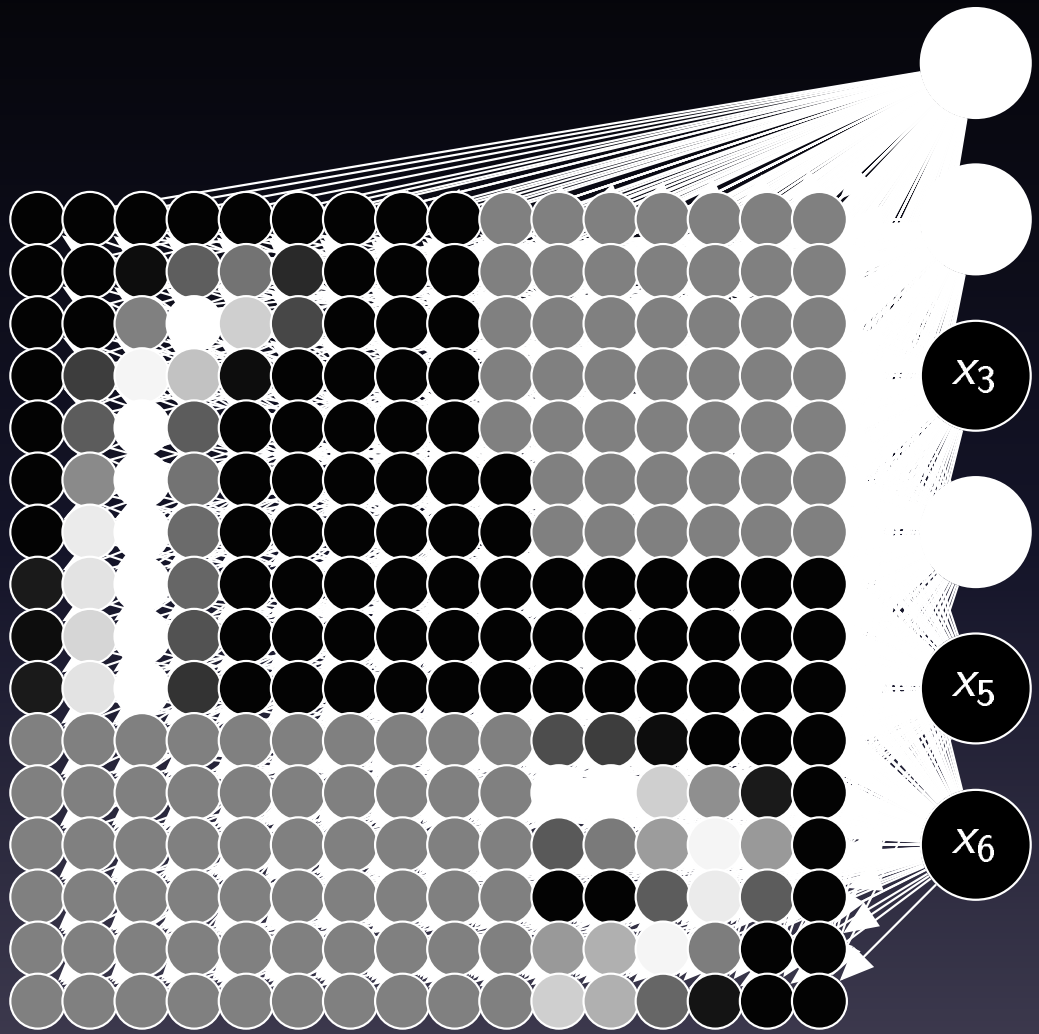

x_1
x_2
x_4
x_5
x_6

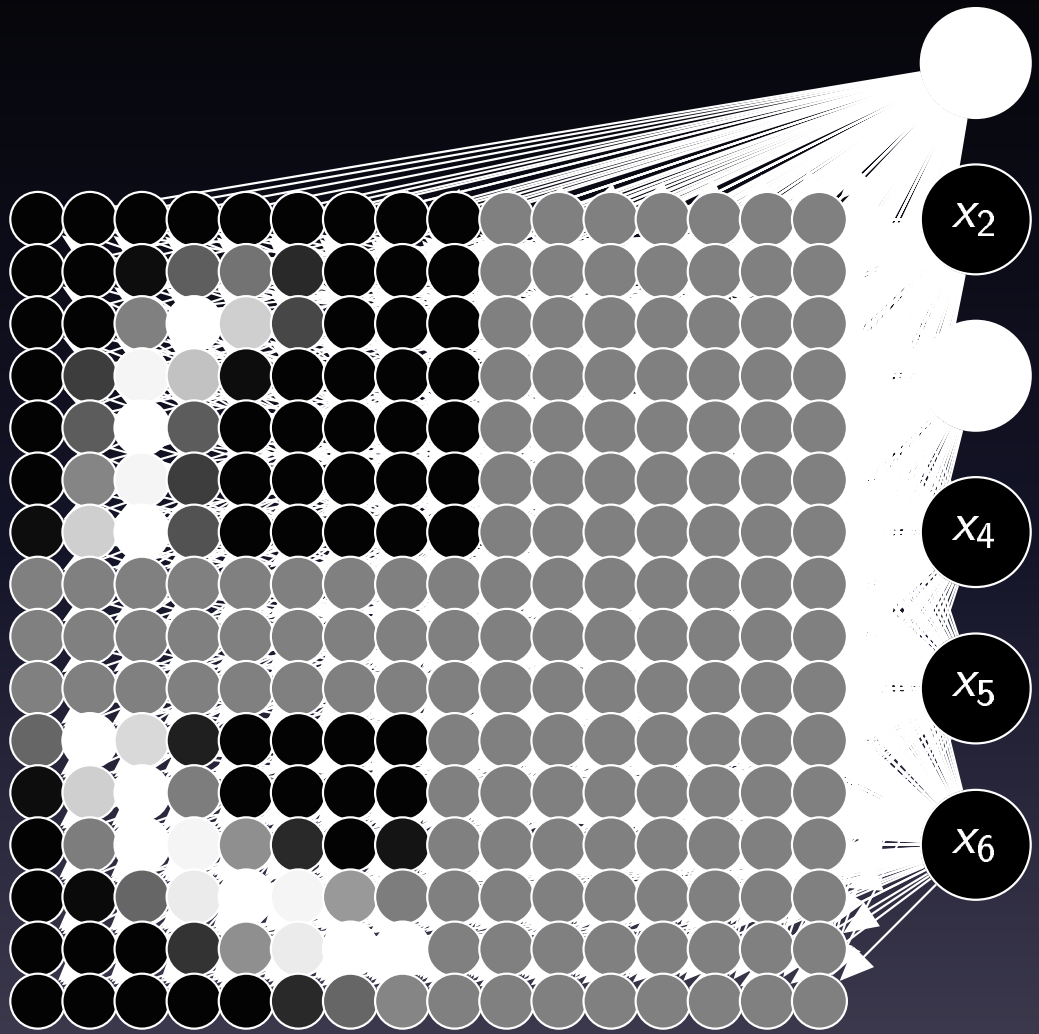

x_1
x_2
x_3
x_5
x_6

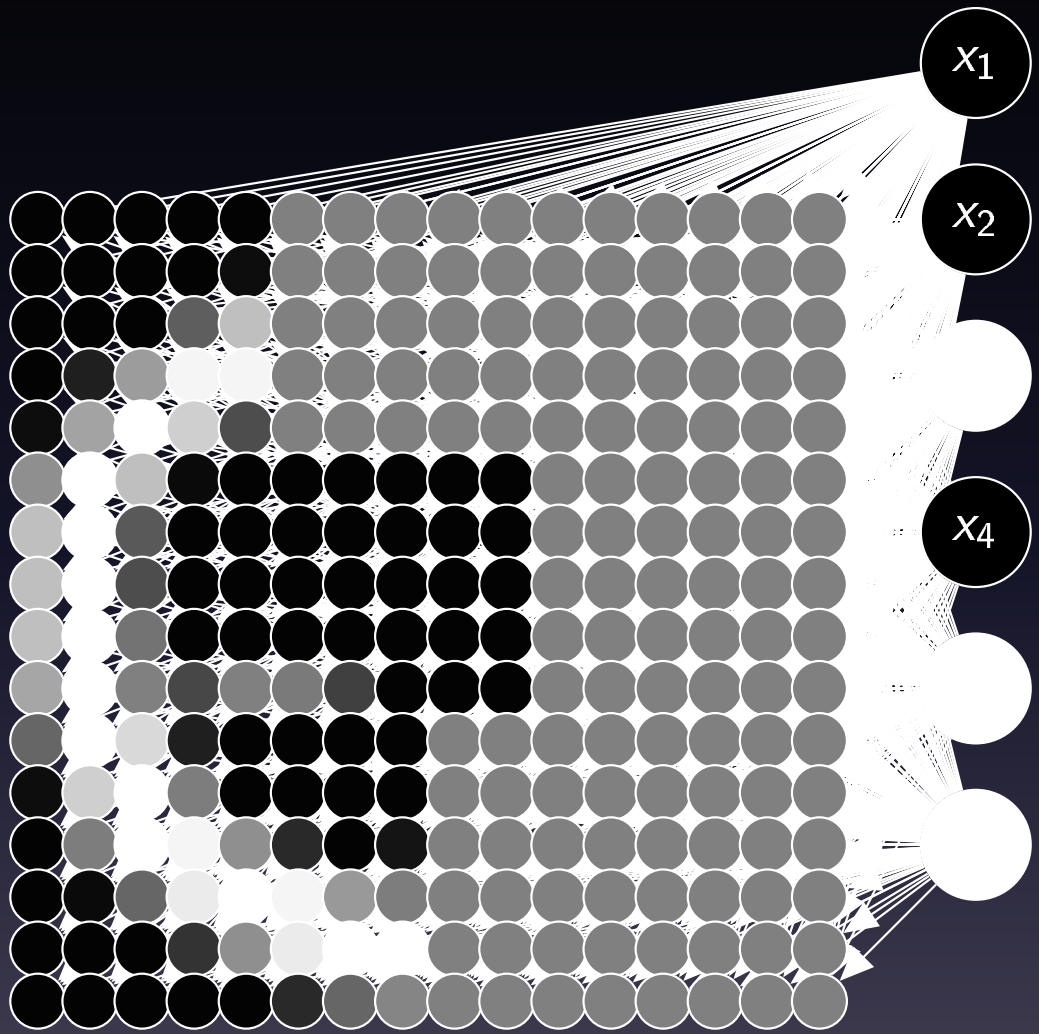

x_1
x_2
x_3
x_4
x_6

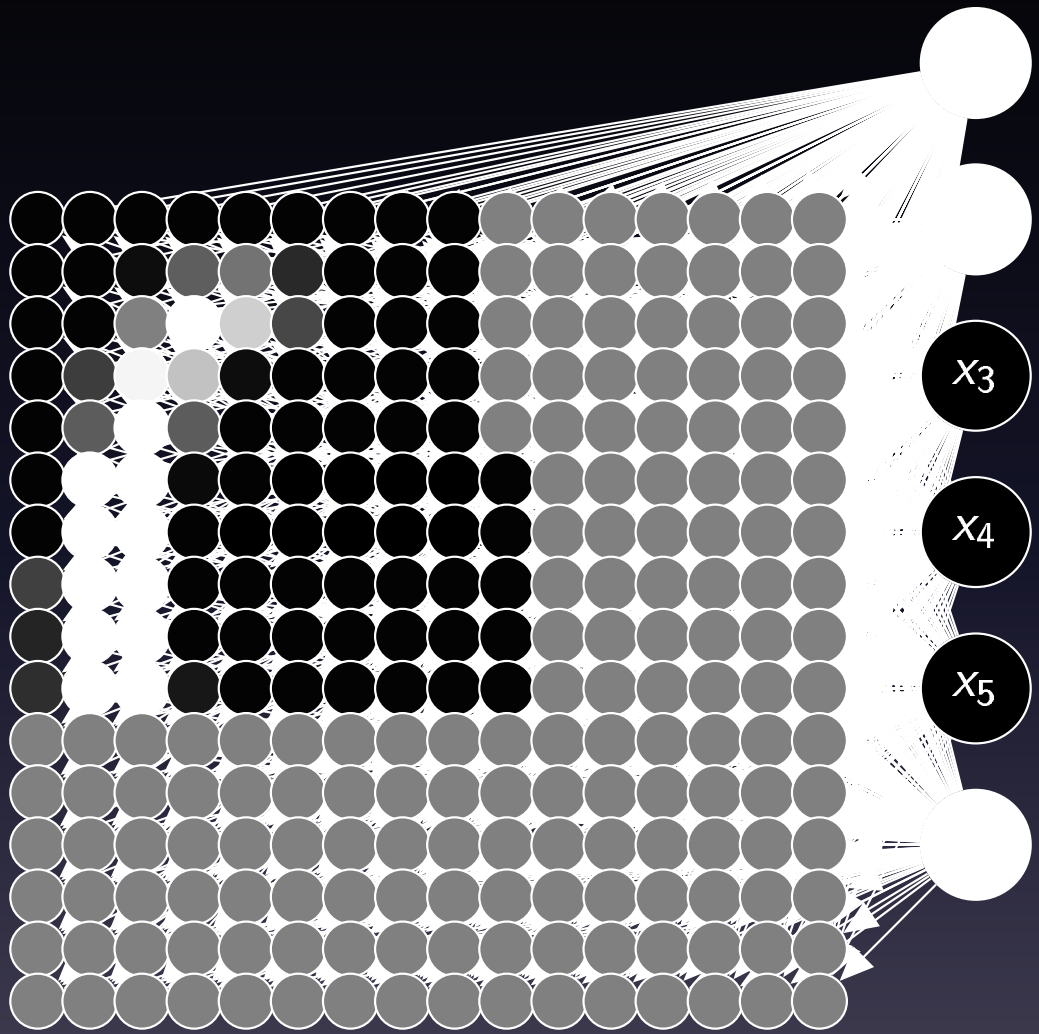

x_1
x_2
x_3
x_4
x_5

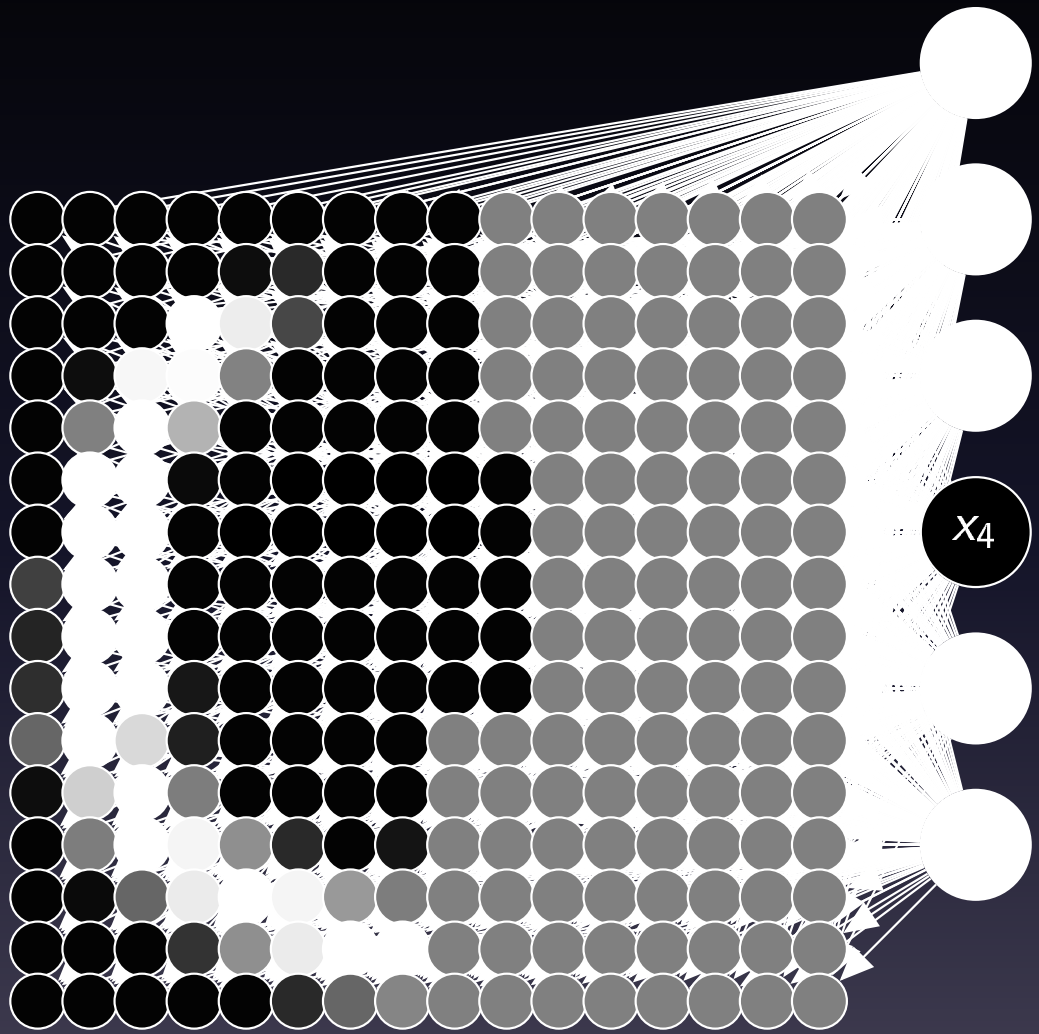

x_1
x_3
x_4
x_5
x_6


x_3
x_5
x_6


x_2
x_4
x_5
x_6


x_1
x_2
x_4


x_3
x_4
x_5


x_4

Restricted Boltzmann Machine

- Represent data, $\mathbf{Y}$, through a set of unobserved latent variables

$$P(\mathbf{Y}) = \sum_{\mathbf{X}} P(\mathbf{Y}|\mathbf{X})P(\mathbf{X}).$$

- Data and latent variables are binary.
- Assume latent variables, $x_{i,j}$, are 'on' with probability π_j.

$$P(\mathbf{X}) = \prod_{i=1}^{n} \prod_{j=1}^{q} \pi_j^{x_{i,j}} (1 - \pi_j)^{(1-x_{i,j})}$$

- Set $\pi_j = \sigma(b_i)$ where $\sigma(\cdot)$ is the sigmoid function[1] and b_i is a 'bias' parameter.

[1] The sigmoid function is $\sigma(z) = \frac{\exp(z)}{1+\exp(z)}$,

Restricted Boltzmann Machine: Binomial Prior

- Parameterizing in this way means

$$P(\mathbf{X}) \propto \prod_{i=1}^{n} \exp\left(\mathbf{x}_{i,:}^{\top}\mathbf{b}\right)$$

which, because $\mathbf{X}$ is binary, is equivalent to

$$P(\mathbf{X}) \propto \prod_{i=1}^{n} \exp\left(\mathbf{x}_{i,:}^{\top}\text{diag}(\mathbf{b})\mathbf{x}_{i,:}\right)$$

Restricted Boltzmann Machine

- Assume a linear-logistic relationship of the form

$$P(\mathbf{y}_{i,j}) = p_{i,j}^{y_{i,j}}(1 - p_{i,j})^{(1-y_{i,j})}$$

  where $p_{i,j}$ is the probability that $y_{i,j} = 1$.
- For RBM it is often given by

$$p_{i,j} = \sigma\left(\mathbf{w}_{j,:}^{\top}\mathbf{x}_{i,:} + c_j\right)$$

- For convenience we will reparameterize

$$p_{i,j} = \sigma\left(c_j\left[\mathbf{w}_{j,:}^{\top}\mathbf{x}_{i,:} - 1\right]\right)$$

Restricted Boltzmann Machine

- Parameterizing in this way implies

$$P(\mathbf{Y}|\mathbf{X},\mathbf{W}) \propto \prod_{i=1}^{n} \exp\left(\mathbf{y}_{i,:}^{\top}\mathrm{diag}(\mathbf{c})\left(\mathbf{W}\mathbf{x}_{i,:}-\mathbf{1}\right)\right)$$

  which, because $\mathbf{Y}$ is binary can be rewritten[2] as

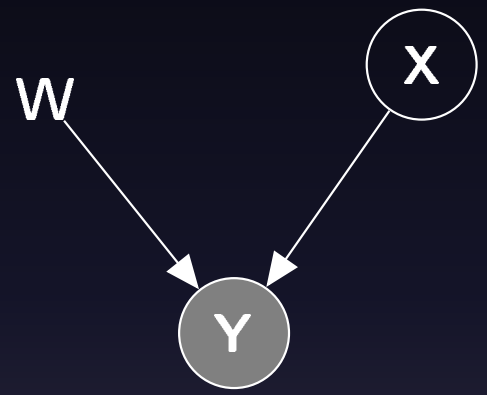
$$P(\mathbf{Y}|\mathbf{X},\mathbf{W}) \propto \prod_{i=1}^{n} \exp\left(-\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)^{\top}\mathrm{diag}(\mathbf{c})\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

[2] To complete the square where we extracted $\mathbf{x}_{i,:}^{\top}\mathbf{W}^{\top}\mathrm{diag}(\mathbf{c})\mathbf{W}\mathbf{x}_{i,:}$ from the constant of proportionality.

Restricted Boltzman Machine

RBM

- Define *linear-logistic relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define binomial prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables* ... ??

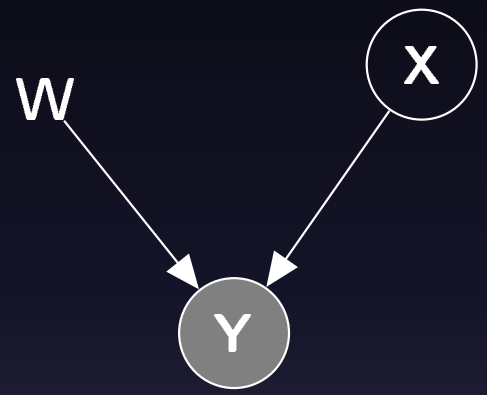


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n}\prod_{j=1}^{p} p_{i,j}^{y_{i,j}}(1-p_{i,j})^{(1-y_{i,j})}$$

Restricted Boltzman Machine

RBM

- Define *linear-logistic relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define binomial prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables* ... ??

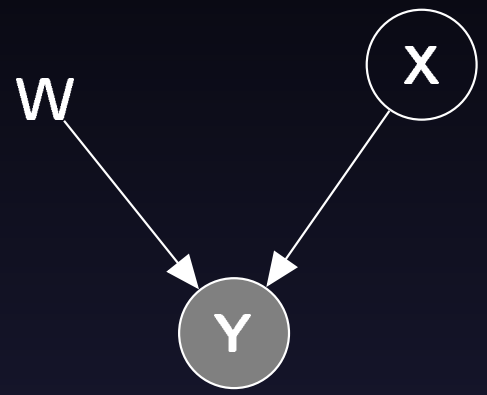


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n}\prod_{j=1}^{p} p_{i,j}^{y_{i,j}}(1-p_{i,j})^{(1-y_{i,j})}$$

Restricted Boltzman Machine

RBM

- Define *linear-logistic relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define binomial prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables* ... ??



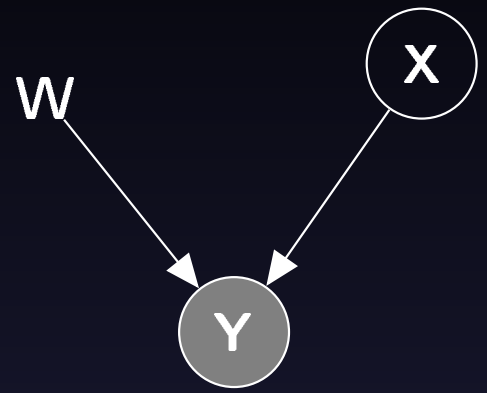
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n}\prod_{j=1}^{p} p_{i,j}^{y_{i,j}}(1-p_{i,j})^{(1-y_{i,j})}$$

$$p(\mathbf{X}) = \prod_{i=1}^{n}\prod_{j=1}^{q} \pi_j^{x_{i,j}}(1-\pi_j)^{(1-x_{i,j})}$$

Restricted Boltzman Machine

RBM

- Define *linear-logistic relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define binomial prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables* ... ??

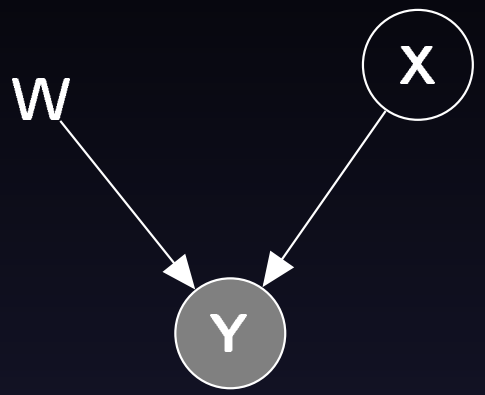


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n}\prod_{j=1}^{p} p_{i,j}^{y_{i,j}}(1-p_{i,j})^{(1-y_{i,j})}$$

$$p(\mathbf{X}) = \prod_{i=1}^{n}\prod_{j=1}^{q} \pi_j^{x_{i,j}}(1-\pi_j)^{(1-x_{i,j})}$$

$$p(\mathbf{Y}|\mathbf{W}) = ??$$

Marginalization of $\mathbf{X}$

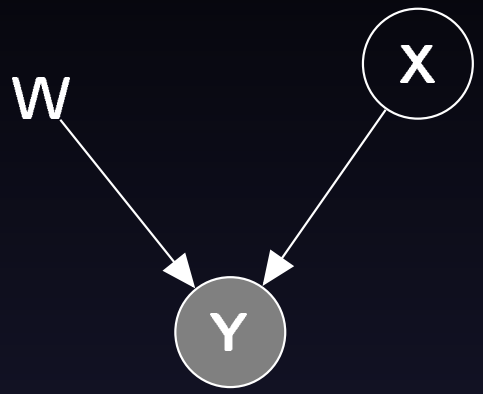


$$P\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) \propto \prod_{i=1}^{n} \exp\left(-\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)^{\top} \operatorname{diag}(\mathbf{c})\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

$$P\left(\mathbf{X}\right) \propto \prod_{i=1}^{n} \exp\left(\mathbf{x}_{i,:}^{\top} \operatorname{diag}\left(\mathbf{b}\right)\mathbf{x}_{i,:}\right)$$

$$P\left(\mathbf{Y}|\mathbf{W}\right) = ??$$

Marginalization of $\mathbf{X}$

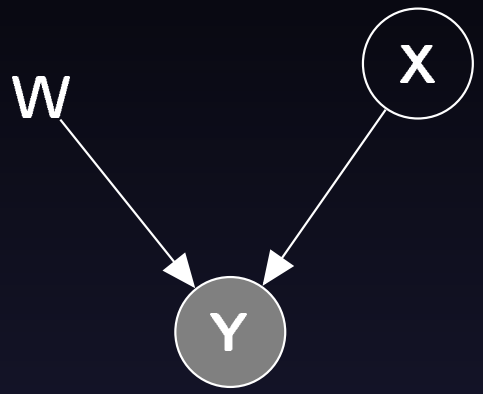


$$P\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) \propto \prod_{i=1}^{n} \exp\left(-\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)^{\top} \operatorname{diag}(\mathbf{c})\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

$$P\left(\mathbf{X}\right) \propto \prod_{i=1}^{n} \exp\left(\mathbf{x}_{i,:}^{\top} \operatorname{diag}\left(\mathbf{b}\right)\mathbf{x}_{i,:}\right)$$

$$P\left(\mathbf{Y}|\mathbf{W}\right) = \sum_{\mathbf{X}} P(\mathbf{Y}|\mathbf{X},\mathbf{W})P(\mathbf{X})$$

Model Factorizes Across Data

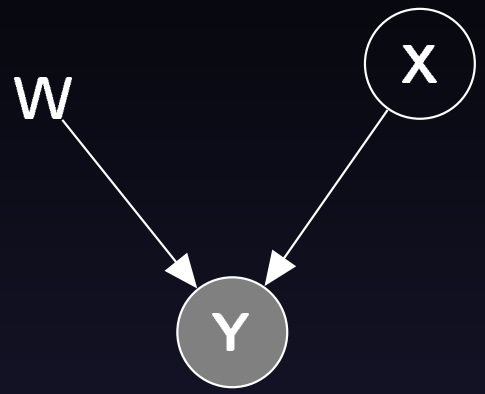


$$P\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:},\mathbf{W}\right) \propto \exp\left(-\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)^{\top}\operatorname{diag}(\mathbf{c})\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

$$P\left(\mathbf{x}_{i,:}\right) \propto \exp\left(\mathbf{x}_{i,:}^{\top}\operatorname{diag}\left(\mathbf{b}\right)\mathbf{x}_{i,:}\right)$$

$$P\left(\mathbf{y}_{i,:}|\mathbf{W}\right) = \sum_{\mathbf{x}_{i,:}} P\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:},\mathbf{W}\right) P\left(\mathbf{x}_{i,:}\right)$$

Model Factorizes Across Data



$$P\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:}, \mathbf{W}\right) \propto \exp\left(-\left(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:}\right)^{\top} \mathrm{diag}(\mathbf{c}) \left(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

$$P\left(\mathbf{x}_{i,:}\right) \propto \exp\left(\mathbf{x}_{i,:}^{\top} \mathrm{diag}\left(\mathbf{b}\right) \mathbf{x}_{i,:}\right)$$

$$P\left(\mathbf{y}_{i,:}|\mathbf{W}\right) = \sum_{\mathbf{x}_{i,:}} P\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:}, \mathbf{W}\right) P\left(\mathbf{x}_{i,:}\right)$$

Unfortunately this sum still contains 2^q terms.

Linear Dimensionality Reduction

Linear Latent Variable Model

- Represent data, $\mathbf{Y}$, with a lower dimensional set of latent variables $\mathbf{X}$.
- Assume a linear relationship of the form

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:},$$

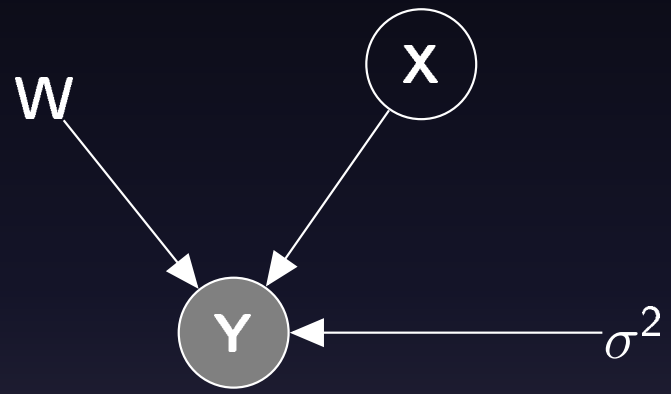
where

$$\boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right).$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.

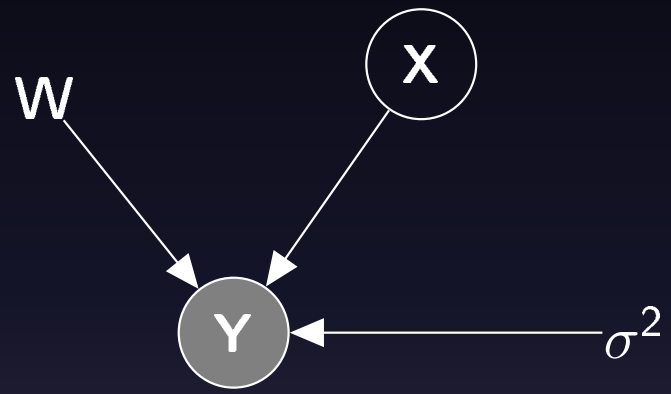


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.

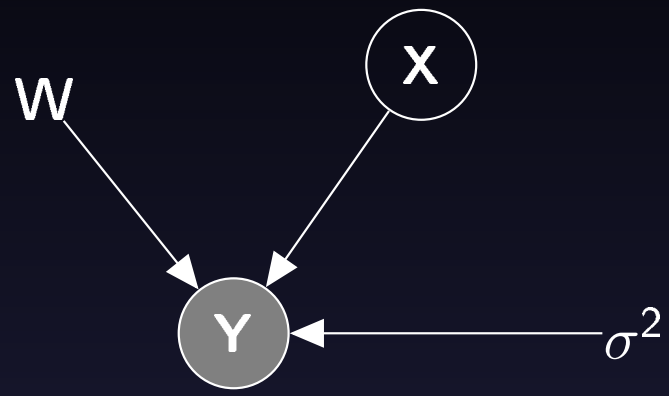


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



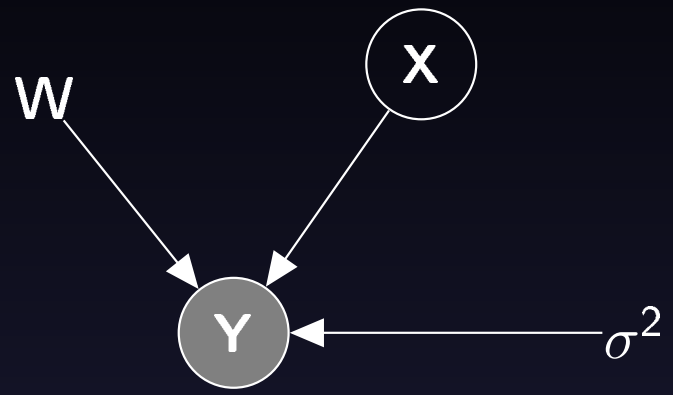
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



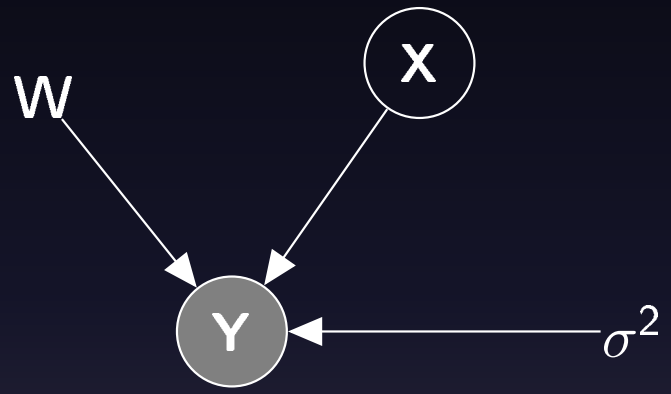
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)



$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

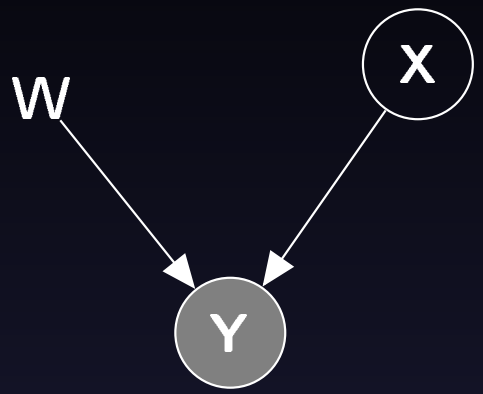
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Relation between RBM and PCA/FA

- RBM is PCA with latent variables and data variables restricted binary.
- Binary restriction means latent features combine in a non-linear way.
- In PCA latent features always combine in a linear way.

PCA and RBM

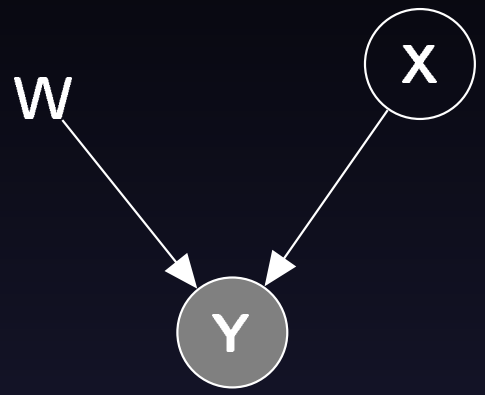


$$P\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:},\mathbf{W}\right) \propto \exp\left(-\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)^{\top}\operatorname{diag}(\mathbf{c})\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

$$P\left(\mathbf{x}_{i,:}\right) \propto \exp\left(\mathbf{x}_{i,:}^{\top}\operatorname{diag}\left(\mathbf{b}\right)\mathbf{x}_{i,:}\right)$$

$$P\left(\mathbf{y}_{i,:}|\mathbf{W}\right) = \sum_{\mathbf{x}_{i,:}} P\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:},\mathbf{W}\right) P\left(\mathbf{x}_{i,:}\right)$$

PCA and RBM

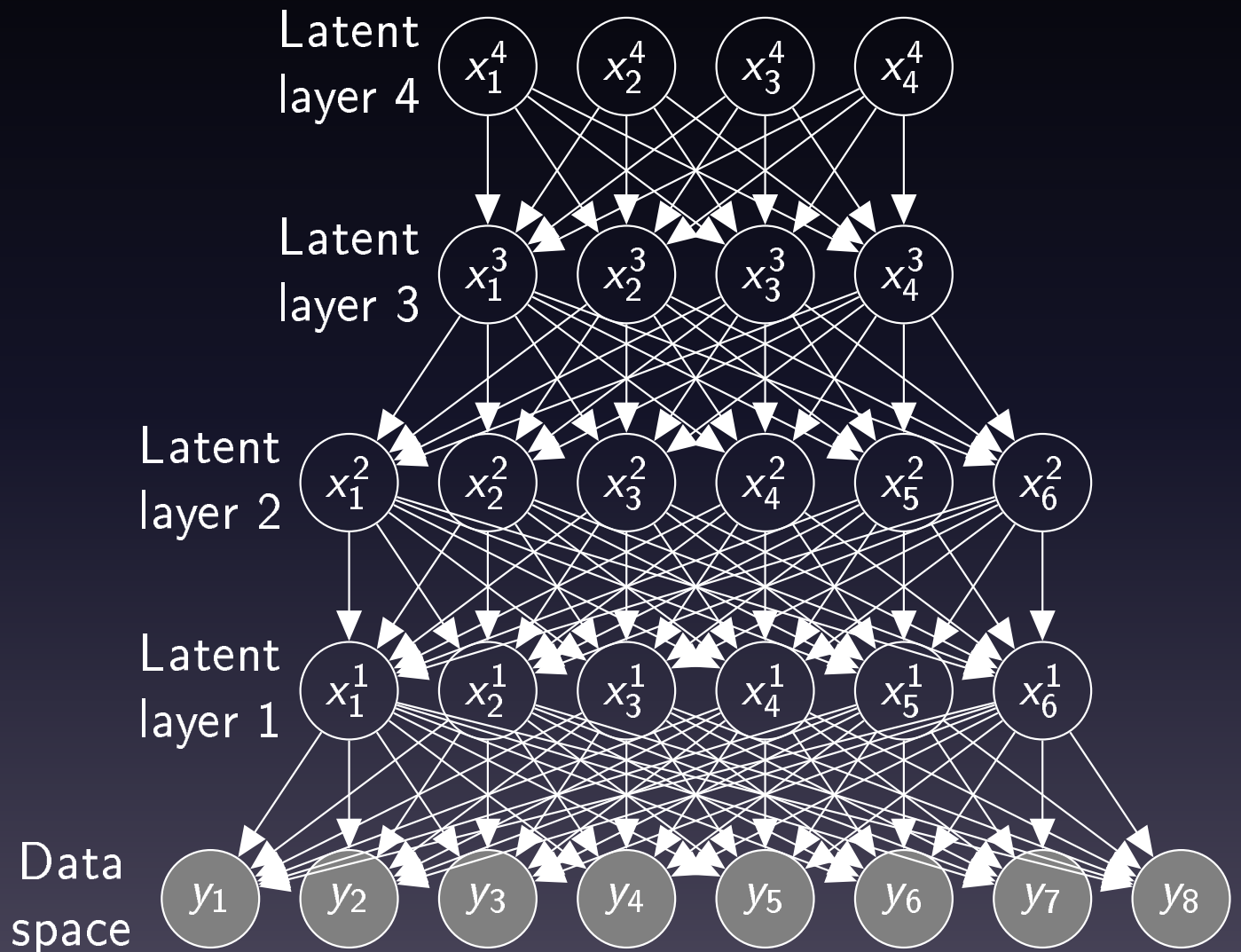


$$p\left(\mathbf{y}_{i,:}|\mathbf{x}_{i,:},\mathbf{W}\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)^\top\left(\mathbf{y}_{i,:}-\mathbf{W}\mathbf{x}_{i,:}\right)\right)$$

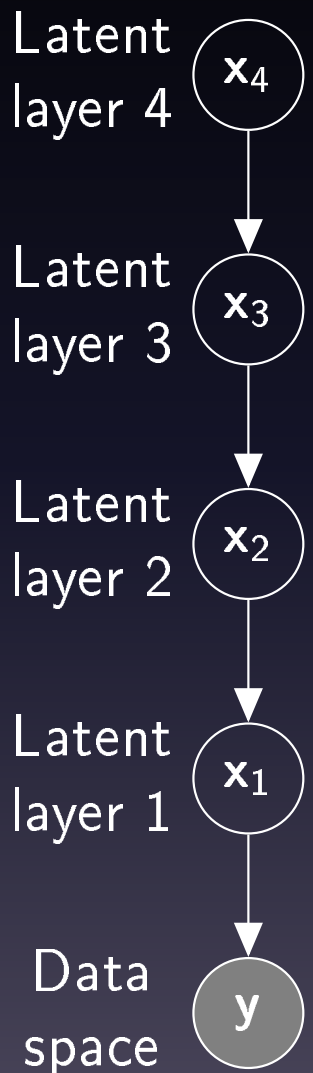
$$p\left(\mathbf{x}_{i,:}\right) \propto \exp\left(-\frac{1}{2}\mathbf{x}_{i,:}^\top\mathbf{x}_{i,:}\right)$$

$$p\left(\mathbf{y}_{i,:}|\mathbf{W}\right) = \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0},\mathbf{W}\mathbf{W}^\top+\sigma^2\mathbf{I}\right)$$

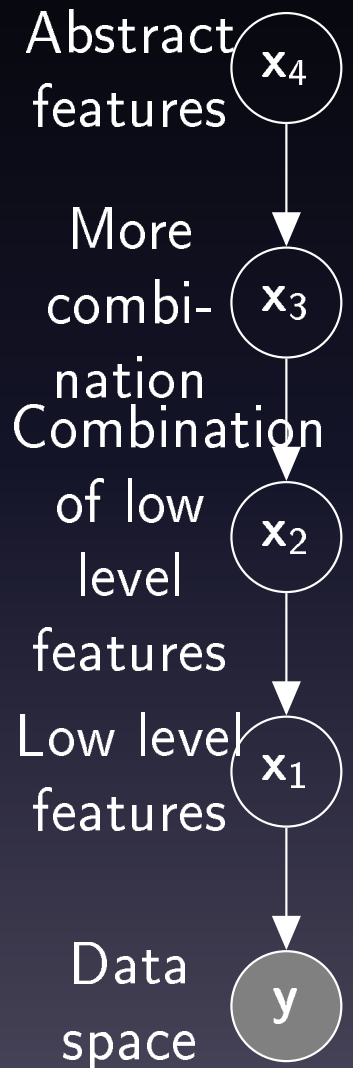
Deep Models


Latent layer 4
Latent layer 3
Latent layer 2
Latent layer 1
Data space
x_1^4 x_2^4 x_3^4 x_4^4
x_1^3 x_2^3 x_3^3 x_4^3
x_1^2 x_2^2 x_3^2 x_4^2 x_5^2 x_6^2
x_1^1 x_2^1 x_3^1 x_4^1 x_5^1 x_6^1
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8

Deep Models


Latent layer 4
$\mathbf{x}_4$
Latent layer 3
$\mathbf{x}_3$
Latent layer 2
$\mathbf{x}_2$
Latent layer 1
$\mathbf{x}_1$
Data space
$\mathbf{y}$

Deep Models


Abstract features
$\mathbf{x}_4$
More combi-nation
$\mathbf{x}_3$
Combination of low level features
$\mathbf{x}_2$
Low level features
$\mathbf{x}_1$
Data space
$\mathbf{y}$

Deep Gaussian Processes

Work with Andreas Damianou

- Deep architectures allow abstraction of features (Bengio, 2009; Hinton and Osindero, 2006; Salakhutdinov and Murray, 2008).
- We use variational approach to stack GP models.

Deep GPs

- Stacking PPCA still leads to a linear latent variable model.
- To stack latent variable models, need a non-linear model.
- The GP-LVM is a non-linear latent variable model.
- Stacking GP-LVM leads to hierarchical GP-LVM.

Hierarchical GP-LVM

(Lawrence and Moore, 2007)

Stacking Gaussian Processes

- Regressive dynamics provides a simple hierarchy.
    - The input space of the GP is governed by another GP.
- By stacking GPs we can consider more complex hierarchies.
- Ideally we should marginalise latent spaces
    - In practice we seek MAP solutions.

Two Correlated Subjects

(Lawrence and Moore, 2007)

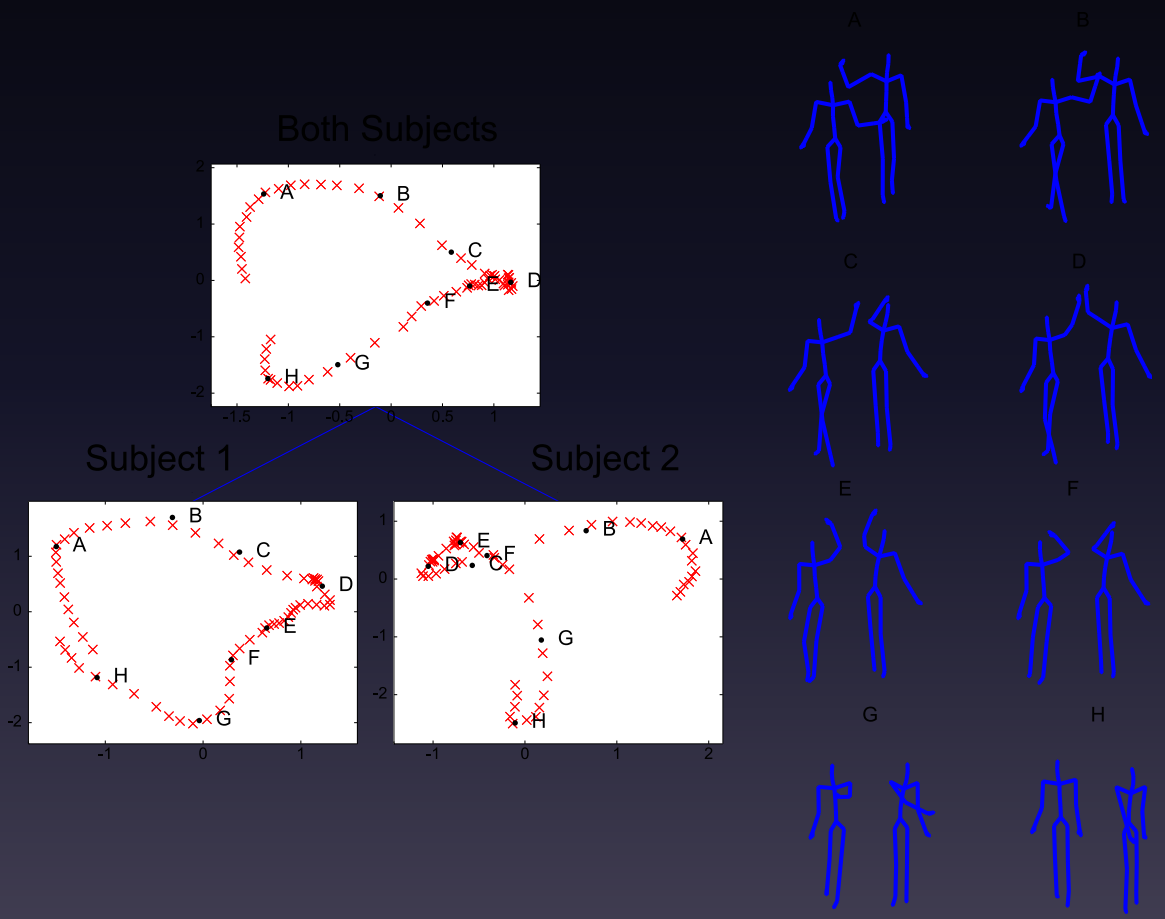


Figure: Hierarchical model of a 'high five'.

Within Subject Hierarchy

(Lawrence and Moore, 2007)

Decomposition of Body

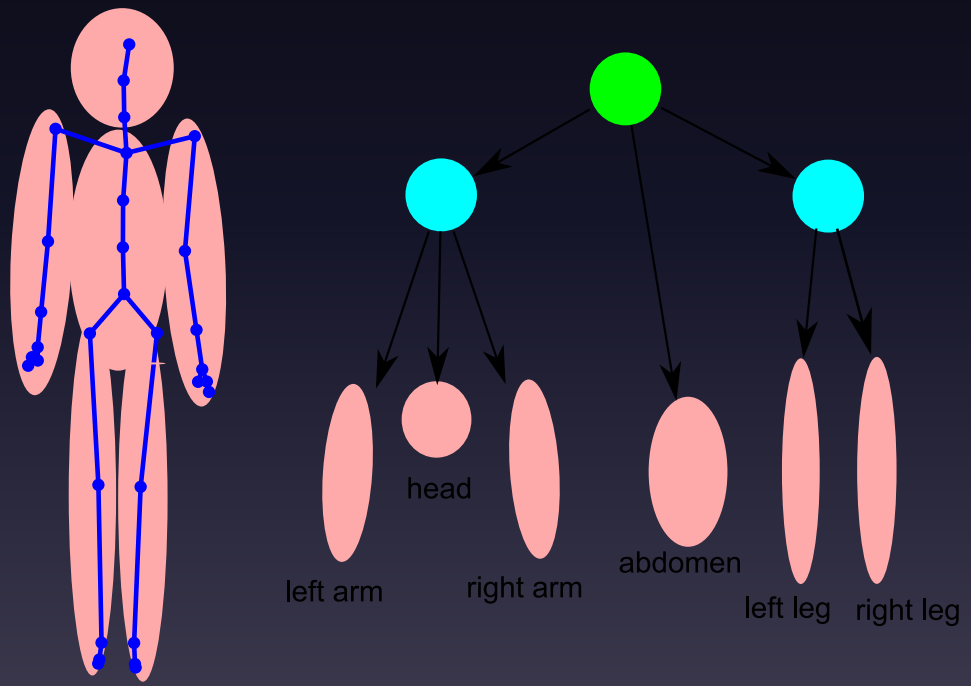


Figure: Decomposition of a subject.

Single Subject Run/Walk

(Lawrence and Moore, 2007)

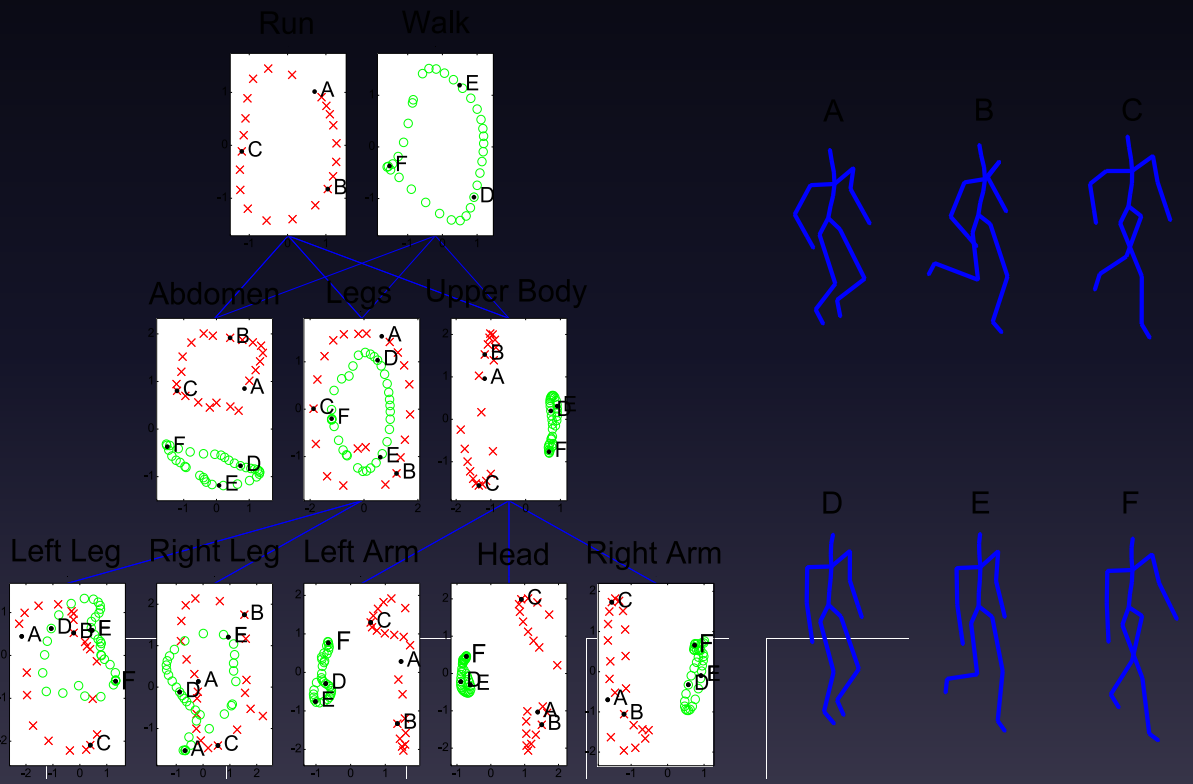
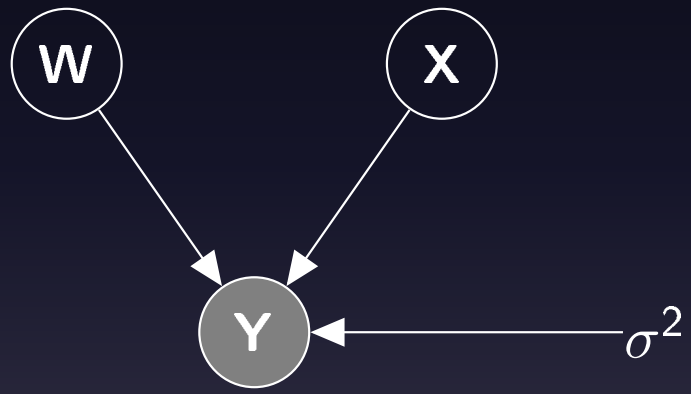


Figure: Hierarchical model of a walk and a run.

▸ Return

Bayesian GP-LVM

- Bayesian GP-LVM allows variational marginalization of $\mathbf{X}$ and $\mathbf{W}$.



- This leads to a Bayesian model where latent dimensionality can be learnt.

Selecting Data Dimensionality

- GP-LVM Provides probabilistic non-linear dimensionality reduction.
- How to select the dimensionality?
- Need to estimate marginal likelihood.
- In standard GP-LVM it increases with increasing q.

Variational Latent Variables

- Variational marginalizing of $\mathbf{X}$ is *also* analytic.
- Need to assume Gaussian $q(\mathbf{X})$.
- Compute expectations of $q(\mathbf{X})$ then analytically marginalize $p(\mathbf{u})$ as before. (Titsias and Lawrence, 2010; Hensman et al., 2012)
  - Requires expectations of $\mathbf{K}_{\mathbf{f},\mathbf{u}}$ and $\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{f}}$.

Non-linear $f(\mathbf{x})$

- In linear case equivalence because $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$

$$p(w_i) \sim \mathcal{N}(\mathbf{0}, \alpha_i)$$

- In non linear case, need to scale columns of $\mathbf{X}$ in prior for $f(\mathbf{x})$.
- This implies scaling columns of $\mathbf{X}$ in covariance function

$$k(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}) = \exp\left(-\frac{1}{2}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})^\top \mathbf{A}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})\right)$$

$\mathbf{A}$ is diagonal with elements α_i^2. Now keep prior spherical

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}(\mathbf{x}_{:,j} | \mathbf{0}, \mathbf{I})$$

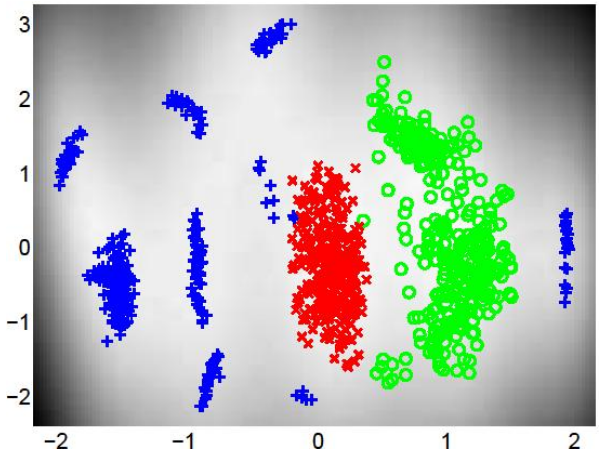
- Covariance functions of this type are known as ARD (see e.g. Neal, 1996; MacKay, 2003; Rasmussen and Williams, 2006).

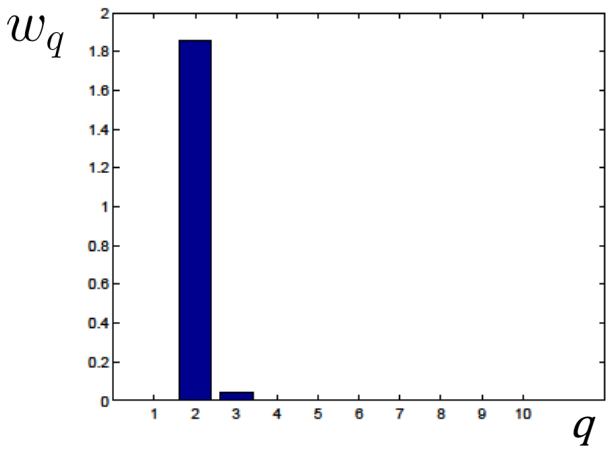
Automatic dimensionality detection

- Achieved by employing an *Automatic Relevance Determination (ARD)* covariance function for the prior on the GP mapping
- $f \sim GP(\mathbf{0}, k_f)$ with

$$k_f(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{i,q} - x_{j,q})^2\right)$$

- Example

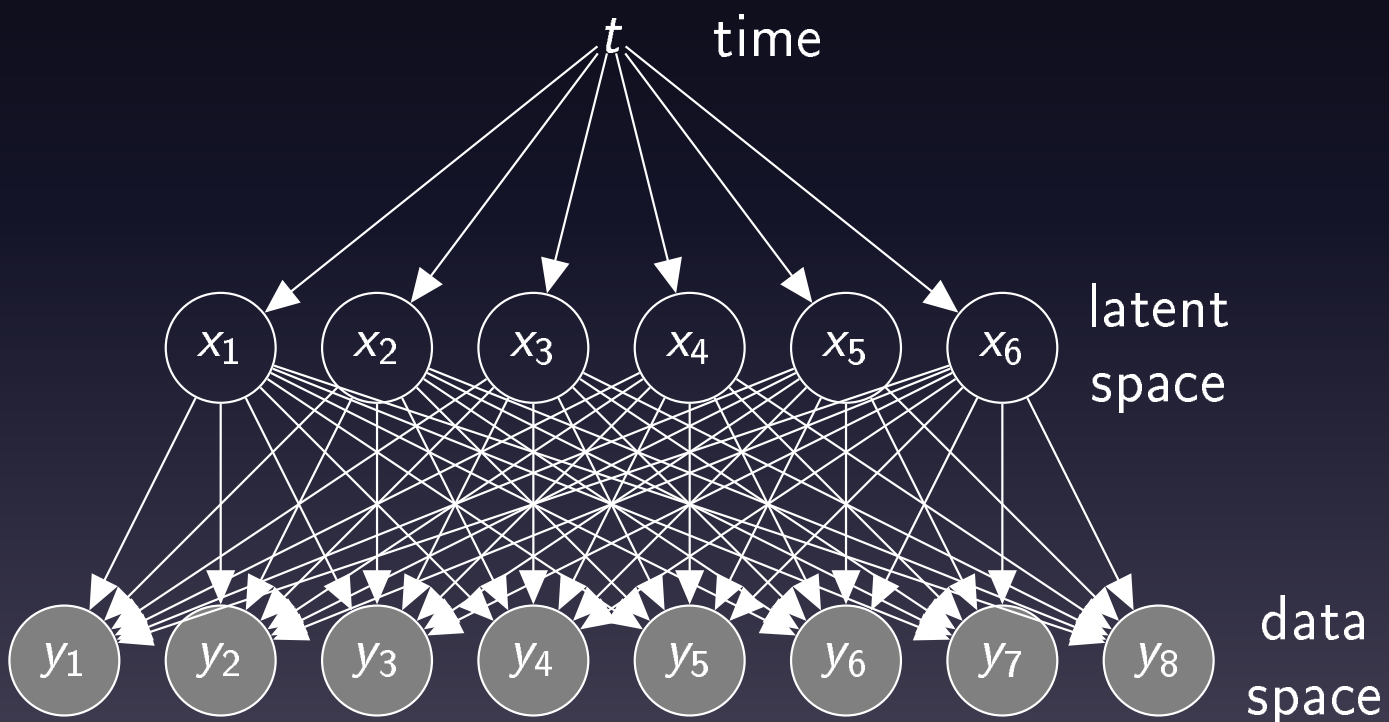




Face Demo

Gaussian Process Dynamical Systems

Work with Andreas Damianou and Michalis Titsias



Gaussian Process over Latent Space

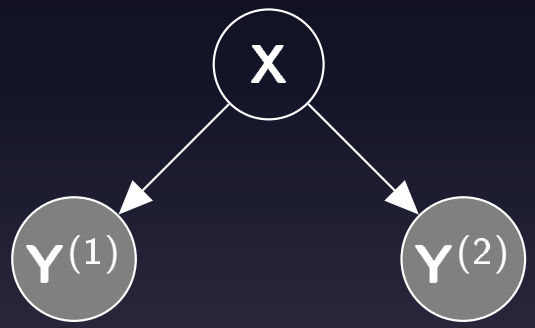
- Assume a GP prior for $p(\mathbf{X})$.
- Input to the process is time, $p(\mathbf{X}|t)$.

Gaussian Process over Latent Space

- Allows to interpret high dimensional video.
- Examples: Missa and Dog Generation.

Modeling Multiple 'Views'

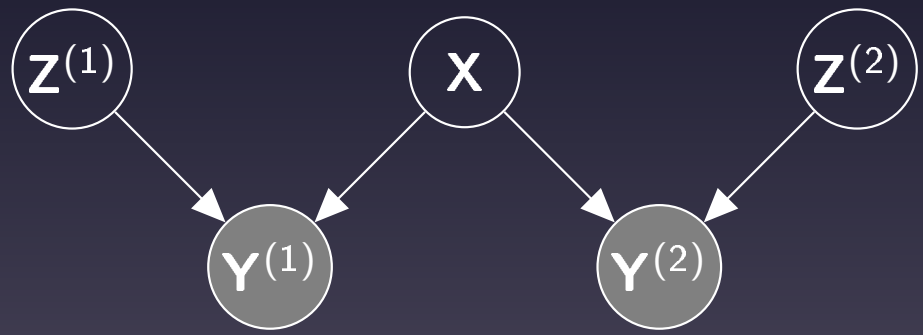
- Single space to model correlations between two different data sources, e.g., images & text, image & pose.
- Shared latent spaces: (Shon et al., 2006; Navaratnam et al., 2007; Ek et al., 2008b)



- Effective when the 'views' are correlated.
- But not all information is shared between both 'views'.
- PCA applied to concatenated data vs CCA applied to data.

Shared-Private Factorization

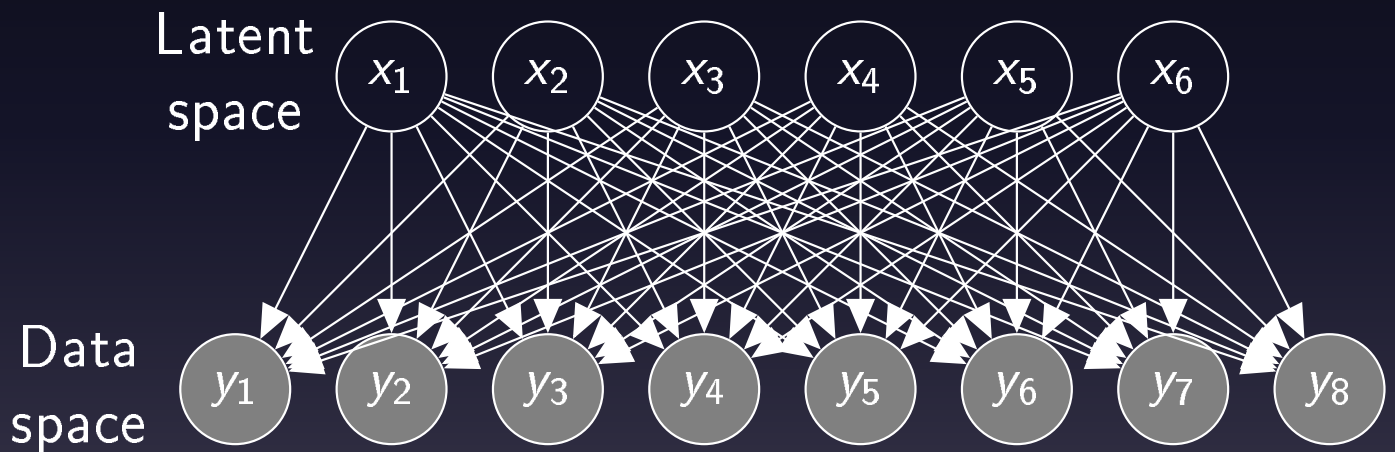
- In real scenarios, the 'views' are neither fully independent, nor fully correlated.
- Shared models
  - either allow information relevant to a single view to be mixed in the shared signal,
  - or are unable to model such private information.
- Solution: Model shared and private information (Virtanen et al., 2011; Ek et al., 2008a; Leen and Fyfe, 2006; Klami and Kaski, 2007, 2008; Tucker, 1958)



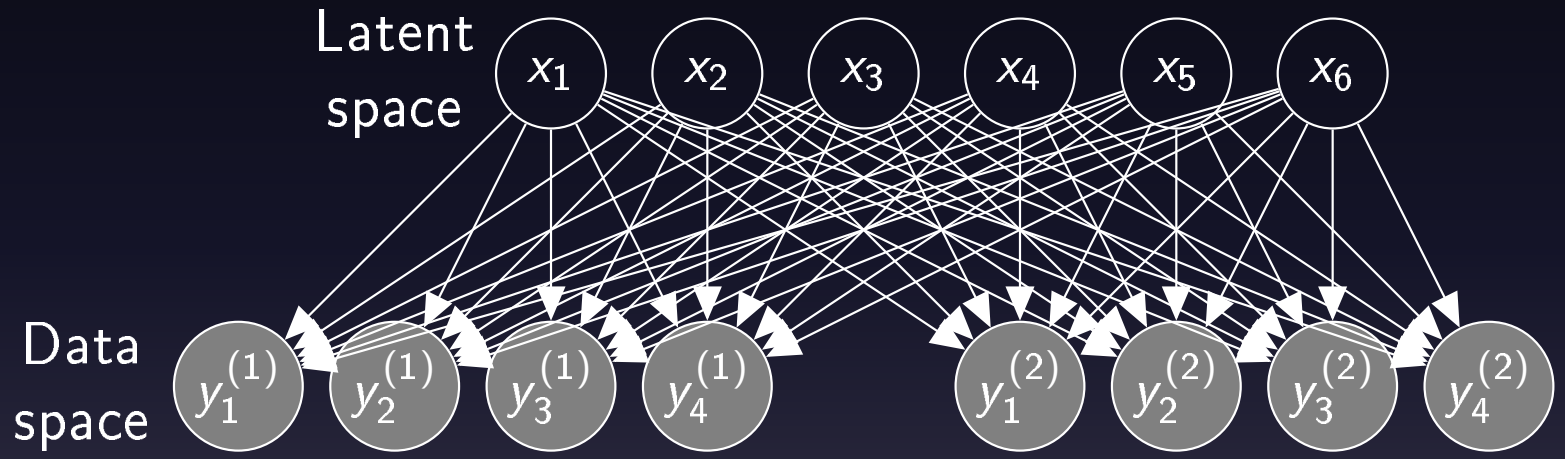
- Probabilistic CCA is case when dimensionality of $\mathbf{Z}$ matches $\mathbf{Y}^{(i)}$ (cf Inter Battery Factor Analysis (Tucker, 1958)).

Manifold Relevance Determination

Work with Andreas Damianou and Carl Henrik Ek



Shared GP-LVM



Separate ARD parameters for mappings to $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$.

Example: Yale faces

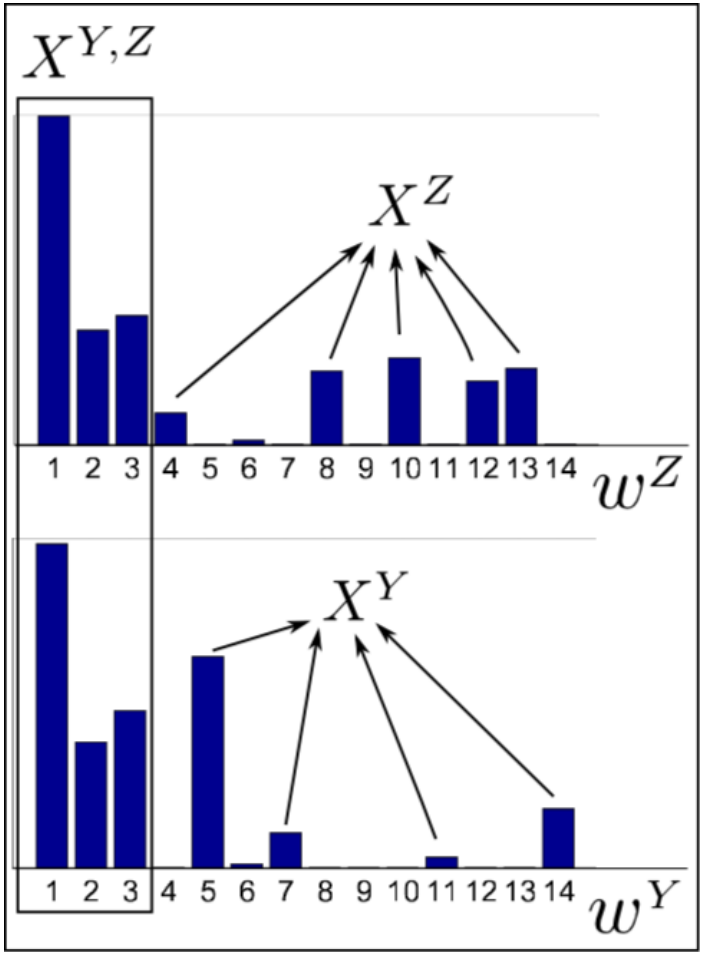
- Dataset Y: 3 persons under all illumination conditions
- Dataset Z: As above for 3 different persons
- Align datapoints $\mathbf{x}_n$ and $\mathbf{z}_n$ only based on the lighting direction

Results

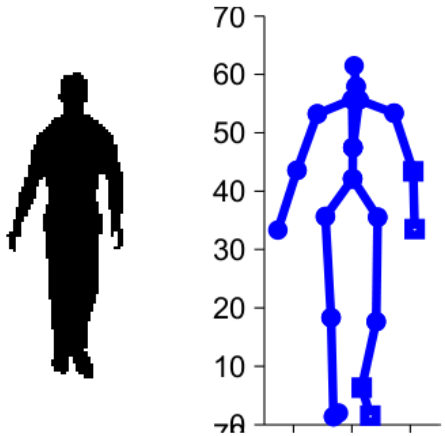
- Latent space X initialised with 14 dimensions

- Weights define a segmentation of X

- Video / demo…



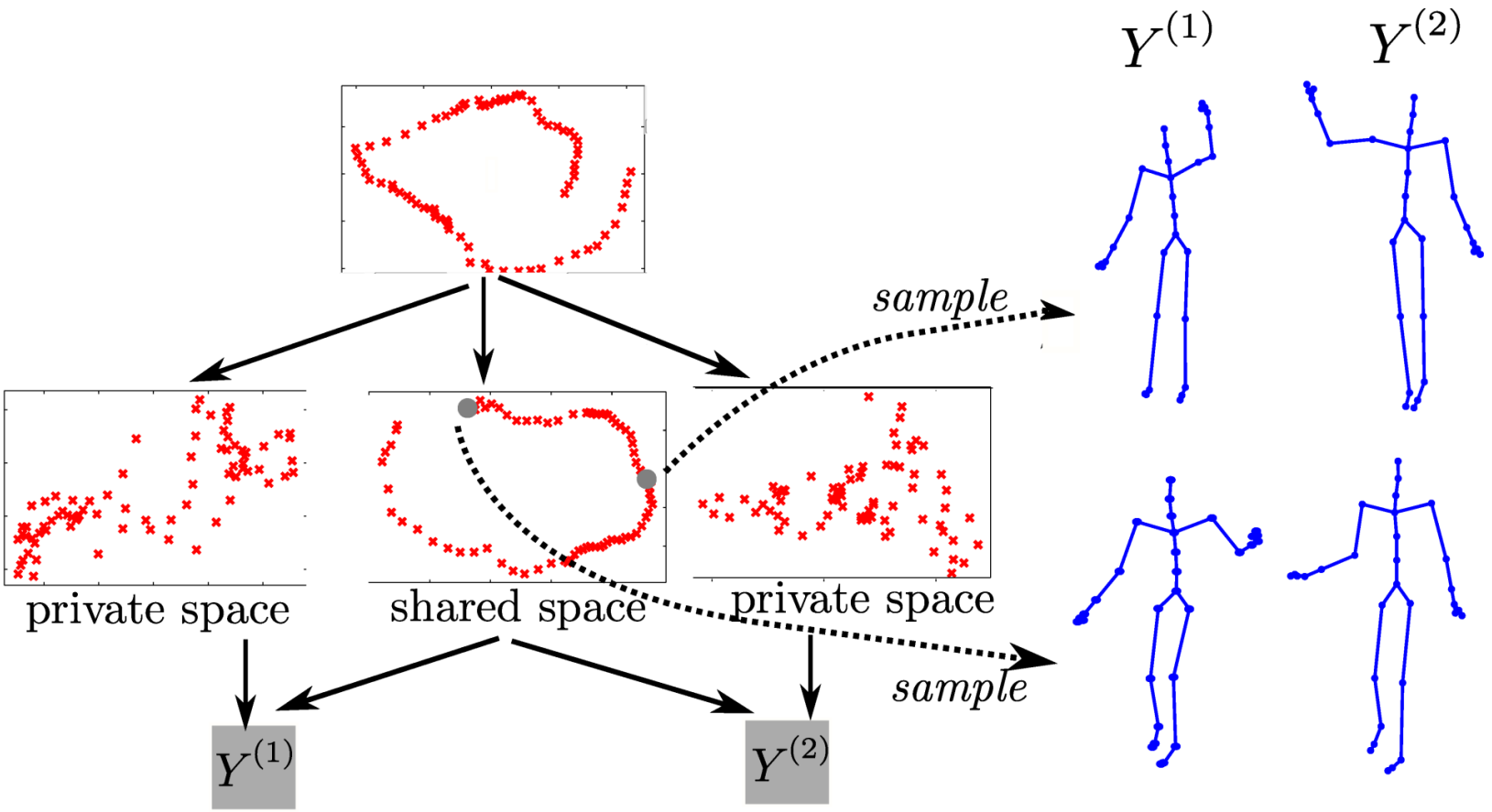
Potential applications..?



Motion Capture

- Revisit 'high five' data.
- This time allow model to learn structure, rather than imposing it.

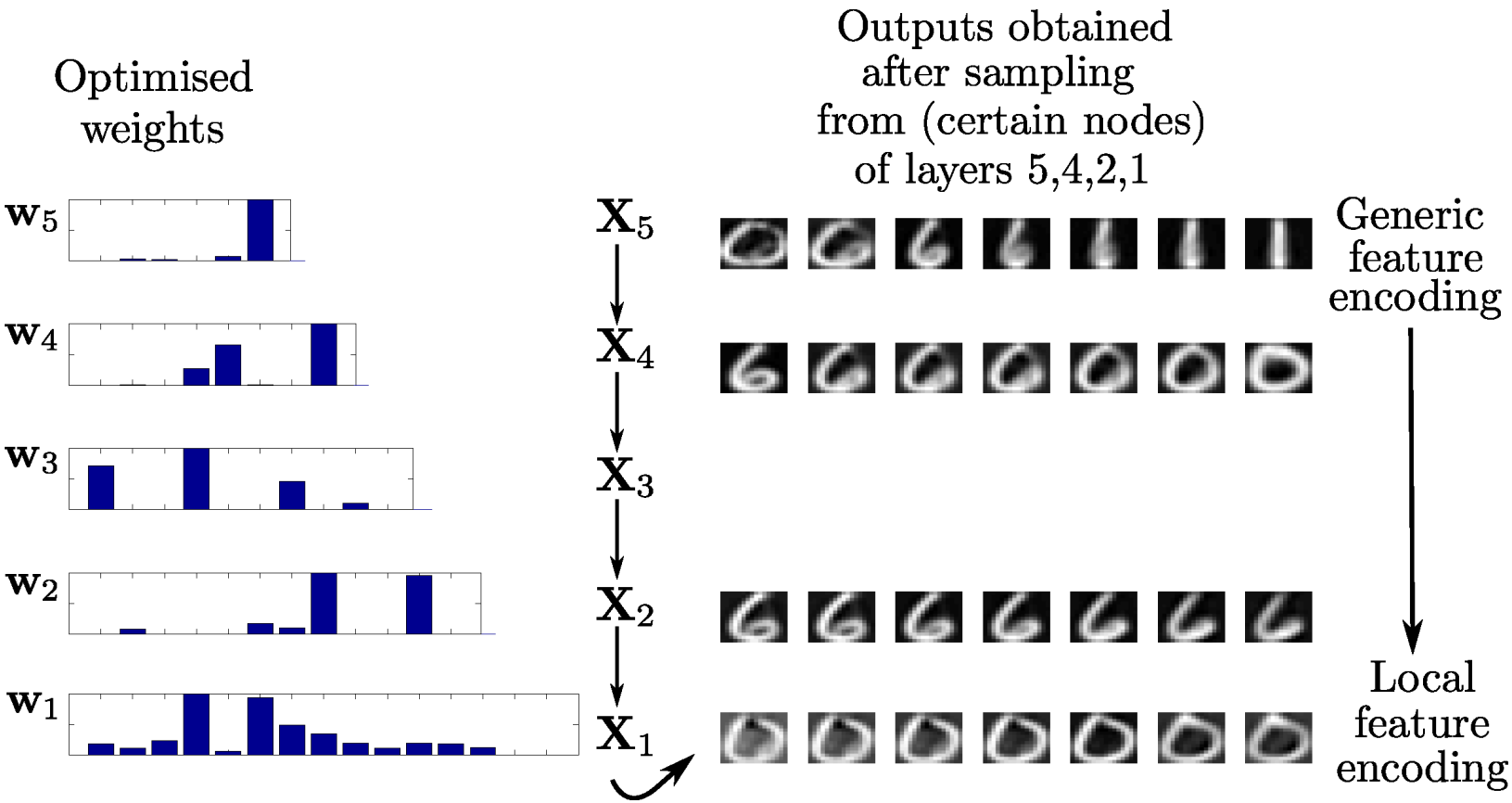
Deep hierarchies – motion capture



Digits Data Set

- Are deep hierarchies justified for small data sets?
- We can lower bound the evidence for different depths.
- For 150 6s, 0s and 1s from MNIST we found at least 5 layers are required.

Deep hierarchies – MNIST



Summary

- Deep models allow abstract representation of data sets at higher levels.
- Deep GPs allow structure learning.
- Current limitation is on data set size.
- Addressing this through work by James Hensman on Stochastic Variational Inference for GPs (NIPS Workshop Poster 'GPs for Big Data').
- Intention is to deploy these models for assimilating a wide range of data types in personalized health (text, survival times, images, genotype, phenotype).
- Requires population scale models with millions of features.

References I

Y. Bengio. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009. ISSN 1935-8237. [DOI].

C. H. Ek, J. Rihan, P. Torr, G. Rogez, and N. D. Lawrence. Ambiguity modeling in latent spaces. In A. Popescu-Belis and R. Stiefelhagen, editors, *Machine Learning for Multimodal Interaction (MLMI 2008)*, LNCS, pages 62–73. Springer-Verlag, 28–30 June 2008a. [PDF].

C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In A. Popescu-Belis, S. Renals, and H. Bourlard, editors, *Machine Learning for Multimodal Interaction (MLMI 2007)*, volume 4892 of *LNCS*, pages 132–143, Brno, Czech Republic, 2008b. Springer-Verlag. [PDF].

Z. Ghahramani, editor. *Proceedings of the International Conference in Machine Learning*, volume 24, 2007. Omnipress. [Google Books] .

J. Hensman, M. Rattray, and N. D. Lawrence. Fast variational inference in the exponential family. *NIPS 2012*, 2012.

G. E. Hinton. Products of experts. In *ICANN 99: Ninth international conference on artificial neural networks*, volume 1, pages 1–6. IEE Press, 1999.

G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18: 2006, 2006.

A. Klami and S. Kaski. Local dependent components analysis. In Ghahramani (2007). [Google Books] .

A. Klami and S. Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72:39–46, 2008.

N. D. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. In Ghahramani (2007), pages 481–488. [Google Books] . [PDF].

G. Leen and C. Fyfe. A Gaussian process latent variable model formulation of canonical correlation analysis. Bruges (Belgium), 26-28 April 2006 2006.

D. J. C. MacKay. Introduction to Gaussian Processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, volume 168 of *Series F: Computer and Systems Sciences*, pages 133–166. Springer-Verlag, Berlin, 1998.

References II

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, U.K., 2003. [Google Books] .

R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society Press, 2007.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. Lecture Notes in Statistics 118.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In S. Roweis and A. McCallum, editors, *Proceedings of the International Conference in Machine Learning*, volume 25, pages 872–879. Omnipress, 2008.

A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao. Learning shared latent structure for image synthesis and robotic imitation. In Weiss et al. (2006).

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. [PDF]. [DOI].

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In Y. W. Teh and D. M. Titterington, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 844–851, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9. [PDF].

L. R. Tucker. An inter-battery method of factor analysis. *Psychometrika*, 23(2):111–136, 1958.

S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In L. Getoor and T. Scheffer, editors, *Proceedings of the International Conference in Machine Learning*, volume 28, 2011.

Y. Weiss, B. Schölkopf, and J. C. Platt, editors. *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In J. Langford and J. Pineau, editors, *Proceedings of the International Conference in Machine Learning*, volume 29, San Francisco, CA, 2012. Morgan Kauffman. To appear.