

Deep Gaussian Processes

Learning Abstract Features with Gaussian Process Models

Neil D. Lawrence

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

University of Cambridge, Engineering Department

1st May 2013

Outline

Template Models

Linear Dimensionality Reduction

Deep Models

Conclusions

Scientists See Promise in Deep-Learning Programs



Hao Zhang/The New York Times

A voice recognition program translated a speech given by Richard F. Rashid, Microsoft's top scientist, into Mandarin Chinese.

By JOHN MARKOFF

Published: November 23, 2012

Using an artificial intelligence technique inspired by theories about how the brain recognizes patterns, technology companies are reporting startling gains in fields as diverse as computer vision, speech recognition and the identification of promising new molecules for designing drugs.

The advances have led to widespread enthusiasm among researchers who design software to perform human

点击查看本文中文版。

Connect With

Log in to see what your friends are sharing on rnytimes.com. Privacy Policy | What's This?

Log In With Facebook

What's Popular Now

King Abdullah of Jordan Has Criticism for All Concerned



7 Marines Killed in Nevada Training Exercise



MOST E-MAILED

MOST VIEWED



1. WELL
Lost Sleep Can Lead to Weight Gain



2. THIS LIFE
The Stories That Bind Us



3. WELL
A New Approach to Hip Surgery



4. Unwanted Electronic Gear Rising in Toxic Piles



5. DAVID BROOKS
The Progressive Shift



6. CONTINUING EDUCATION SPECIAL SECTION
A Gray Jobs Market for All Ages



7. Vatican's Bureaucracy Tests Even the Infallible

FACEBOOK

TWITTER

GOOGLE+

SAVE

E-MAIL

SHARE

PRINT



THE NEW YORKER

SUBSCRIBE
and save
up to 88%

- * SUBSCRIBE
- * RENEW
- * GIVE A GIFT
- * INTERNATIONAL ORDERS
- * ONLINE ARCHIVE



- SUBSCRIBE
- THIS WEEK'S ISSUE
- NEWS
- CULTURE
- POLITICS
- BOOKS
- BUSINESS
- CARTOONS
- HUMOR
- ARCHIVE

- DOUBLE TAKE
- PHOTO BOOTH
- DAILY SHOUTS
- PAGE-TURNER
- DAILY COMMENT
- AMY DAVIDSON
- JOHN CASSIDY
- BOROWITZ
- RICHARD BRODY

THE NEW YORKER | ONLINE ONLY

NEWS DESK

Reporting the latest on Washington and the world.



« How Susan Rice Sees the World | Main | Moral Machines »

NOVEMBER 25, 2012

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

POSTED BY GARY MARCUS

- Share 677
- Tweet 361
- +1
- PRINT
- MORE
- COMMENTS

Can a new technique known as deep learning revolutionize artificial intelligence, as yesterday's [front-page article](#) at the *New York Times* suggests?

There is good reason to be excited about deep learning, a sophisticated "machine learning" algorithm that far exceeds many of its predecessors in its abilities to recognize syllables and images. But there's also good reason to be skeptical. While the *Times* reports that "advances in an artificial intelligence technology that can recognize patterns



WELCOME

SIGN IN | HELP | REGISTER

Search Web site Find

MOST POPULAR MOST E-MAILED THIS ISSUE

1. Andy Borowitz: Cheney Marks Tenth Anniversary of Pretending There Was Reason to Invade Iraq
2. Amy Davidson: Life After Steubenville
3. William Finnegan: Gina Rinehart, Australia's Mining Billionaire
4. Maria Bustillos: On Video Games and Storytelling: An Interview with Tom Bissell
5. Lena Dunham: Lifelong Canine Cravings

THE
NEW YORKER
SUBSCRIBE TODAY!

CLICK
HERE

THE NEW YORKER
DIGITAL



TABLET, MOBILE, AND MORE

Newsletter sign-up: Enter e-mail address Submit

Google To Expand Knowledge Graph Through Hire Of Geoffrey Hinton

Mar 14, 2013 • 8:23 am | (10)

by [Barry Schwartz](#) | Filed Under [Google Search Engine](#)

If I had to place one search priority above all else, I'd say right now, Google's most ambitious project is the [knowledge graph](#). Yea, they are pushing Google+ big time, but the knowledge graph is a level above all of that technically.

Of course, Google has an outstanding team working on this project lead by one of the smartest people I've ever met Amit Singhal.

To take the knowledge graph to the next level, Google has hired/acquired Geoffrey Hinton and his team at DNNresearch. Geoffrey posted a note on his [Google+](#) page about it:



Last summer, I spent several months working with Google's Knowledge team in Mountain View, working with Jeff Dean and an incredible group of scientists and engineers who have a real shot at making spectacular progress in machine learning. Together with two of my recent graduate students, Ilya Sutskever and Alex Krizhevsky (who won the 2012 ImageNet competition), I am betting on Google's team to be the epicenter of future breakthroughs. That means we'll soon be joining Google to work with some of the smartest engineering minds to tackle some of the biggest challenges in computer science. I'll remain part-time at the University of Toronto, where I still have a lot of excellent graduate students, but at Google I will get to see what we can do with very large-scale computation.

I know we just scratched the surface of the knowledge graph and I am excited to see where it takes us in the future.

I am just glad I don't have to figure out how to get us there. I get to just sit and enjoy the ride.

[← PREV STORY](#) [NEXT STORY →](#)

49

10

16

[Tweet](#)

[+1](#)

[Like](#)

[SHARE](#)

SUBSCRIBE



Enter Email Address

Subscribe Now

[SUBSCRIBE OPTIONS >](#)

ADVERTISERS

SEARCH BUZZ VIDEO



Subscribe

Google Panda 25, Next Gen Penguin, DNN



[SUBSCRIBE >](#) [MORE VIDEOS >](#) [VIDEO DETAILS >](#)

ROUNDTABLE SPONSORS

BROWSE BY:

- [> Browse by Date](#)
- [> Find by Category](#)
- [> Discover by Author](#)
- [> Scan Most Recent](#)
- [> See Comments](#)
- [> View Tag Cloud](#)

SEM FORUM THREADS

[WebmasterWorld Forums](#)

ENTERPRISE

research

software

analytics

FOLLOW WIRED
ENTERPRISE

Google Hires Brains that Helped Supercharge Machine Learning

BY ROBERT MCMILLAN 03.13.13 6:30 AM

Follow @bobmcmillan

Like 270

Tweet 218

+1 114

Share 34

MOST RECENT WIRED POSTS



Jawbone's Up
Fitness Band Is Now
Android-Compatible



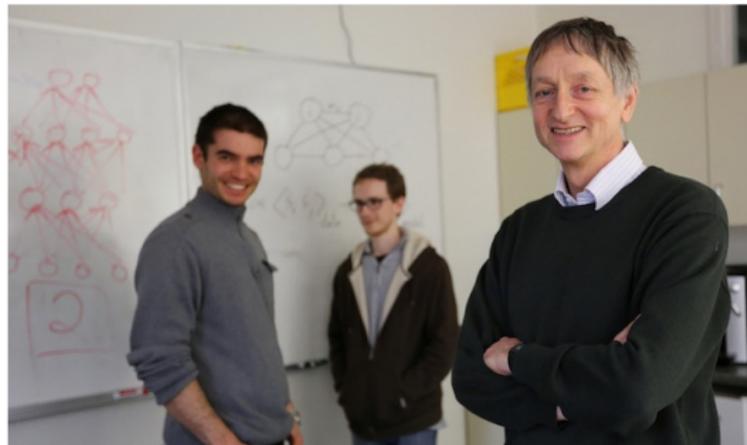
Review: Ecovacs
Winbot, a Window-
Cleaning Robot



Sherlock, Professor
X and Margaery
Tyrell Team for Neil
Gaiman Radio Play



Video: Robo-
Chopper Dives and
Grabs Objects Like a
Bird of Prey



Google Hires Geoffrey Hinton | Google Hires Brains that... | Geoffrey Hinton - Google

https://plus.google.com/102889418997957626067/posts

My Boosters LastPass - Dow... Add to TripIt Proverbi napol... IEEE Xplore - On... Google Maps Other Bookmarks

+Neil Search Images Maps Play YouTube News Gmail Drive Calendar More -

Google+

Home Profile Explore Events Photos Communities Find people Local Games Hangouts More

Geoffrey Hinton  1,734 have him in circles ML People 23 in common

About **Posts** Photos Videos

Geoffrey Hinton 12 Mar 2013 · Public

Last summer, I spent several months working with Google's Knowledge team in Mountain View, working with Jeff Dean and an incredible group of scientists and engineers who have a real shot at making spectacular progress in machine learning. Together with two of my recent graduate students, Ilya Sutskever and Alex Krizhevsky (who won the 2012 ImageNet competition), I am betting on Google's team to be the epicenter of future breakthroughs. That means we'll soon be joining Google to work with some of the smartest engineering minds to tackle some of the biggest challenges in computer science. I'll remain part-time at the University of Toronto, where I still have a lot of excellent graduate students, but at Google I will get to see what we can do with very large-scale computation.

 +418 167

64 comments

Reza Samahin 15 Mar 2013
 +Geoffrey Hinton congrats to you and your team from an old UofT eng grad. Wish I were young again to contribute to your endeavour.

43 IN HIS CIRCLES

-  George Dahl
-  David Reichert
-  Nitish Srivastava
-  Jacqueline Ford
-  Aaron Hertzmann
-  Navdeep Jaitly

23 IN COMMON WITH YOU



1,734 HAVE

direction for further research.

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

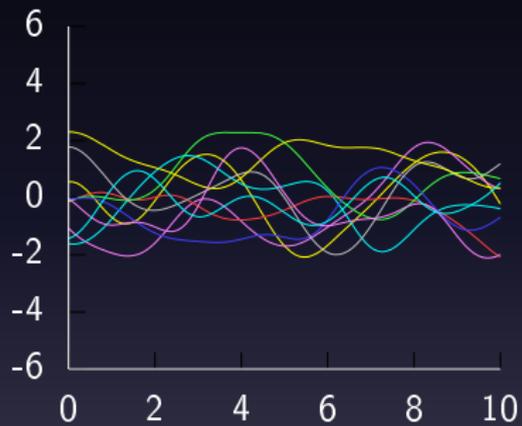
According to the hype of 1987, neural networks were meant to be intelligent models which discovered features and patterns in data. Gaussian processes in contrast are simply smoothing devices. How can Gaussian processes possibly replace neural networks? What is going on?

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.

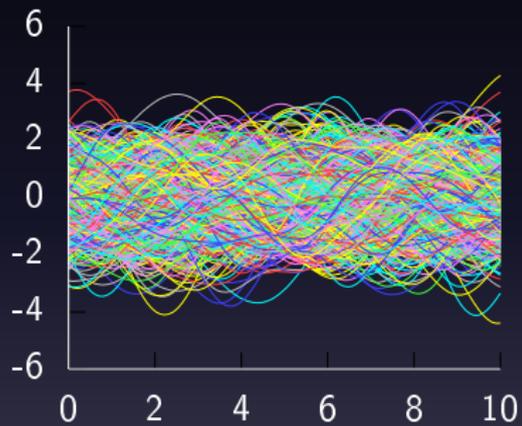
Structure of Priors

MacKay: NIPS Tutorial 1997 “Have we thrown out the baby with the bathwater?” (Published as ?) Also noted by (?)

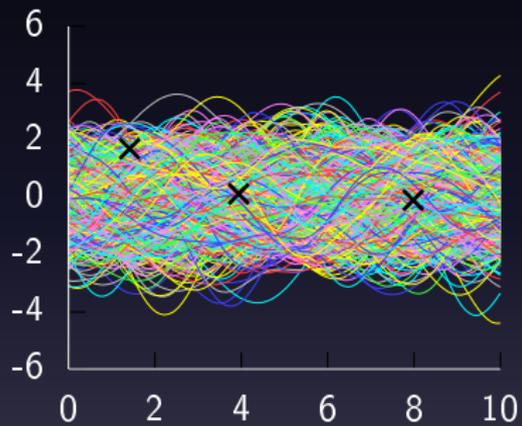
Gaussian Processes



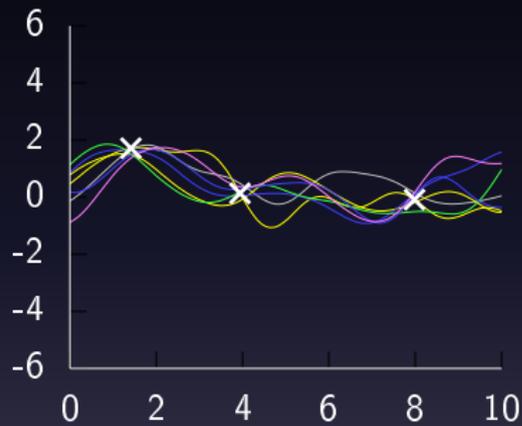
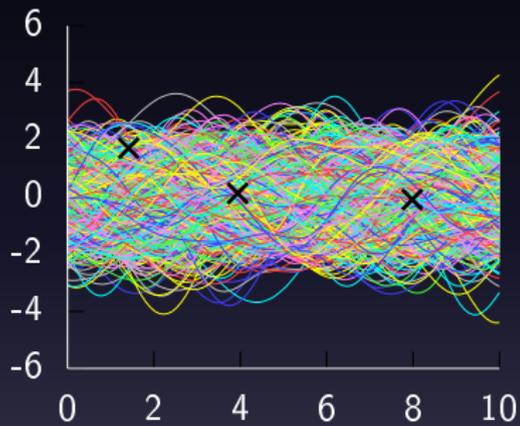
Gaussian Processes



Gaussian Processes



Gaussian Processes



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 Deep Learning

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 Deep Learning

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 Deep Learning

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 Deep Learning

USPS Data Set Handwritten Digit

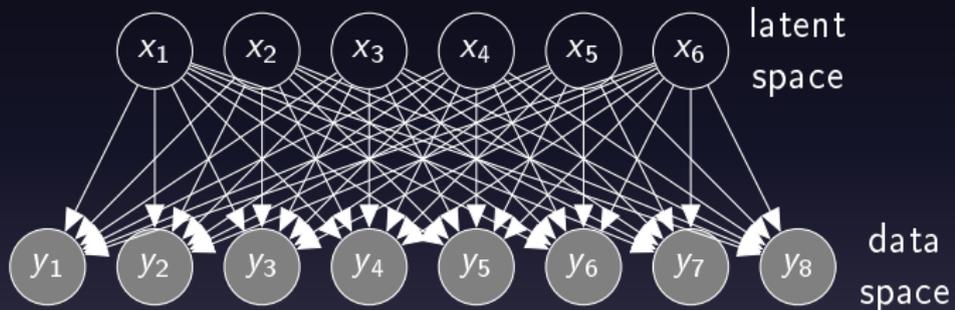
- 3648 Dimensions
 - 64 rows by 57 columns
 - Space contains more than just this digit.
 - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!

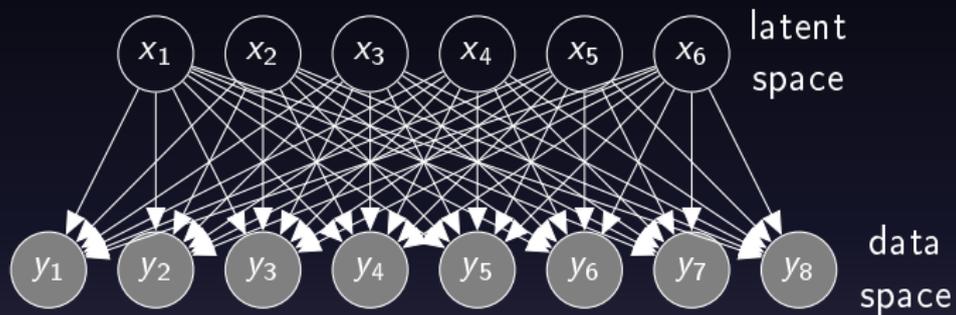


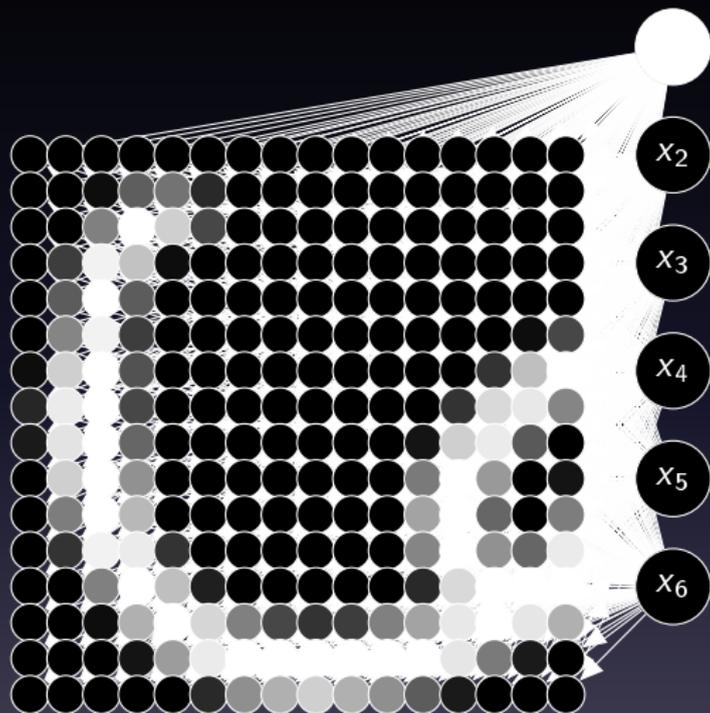
Template Model of Digits

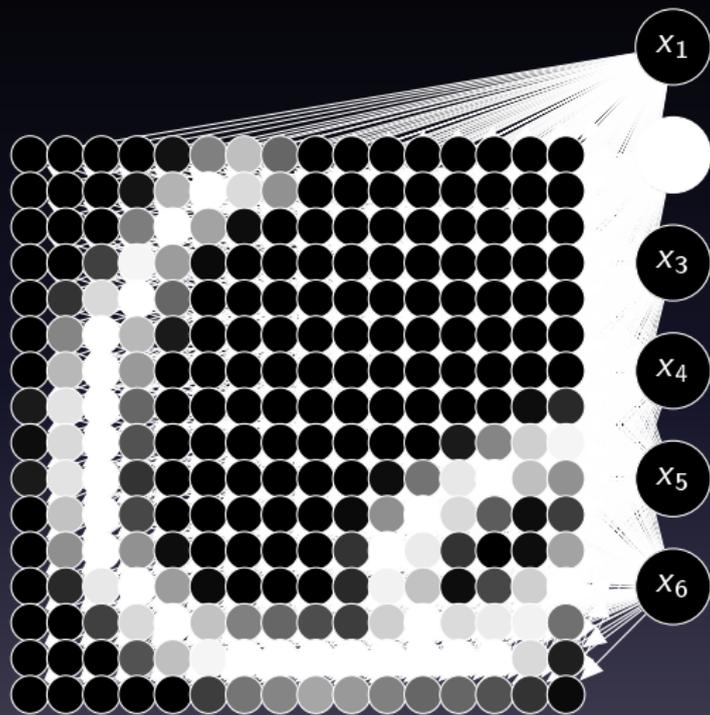
- Design a set of 'latent' features, which generate the 6.
- Global template: memorize data set.

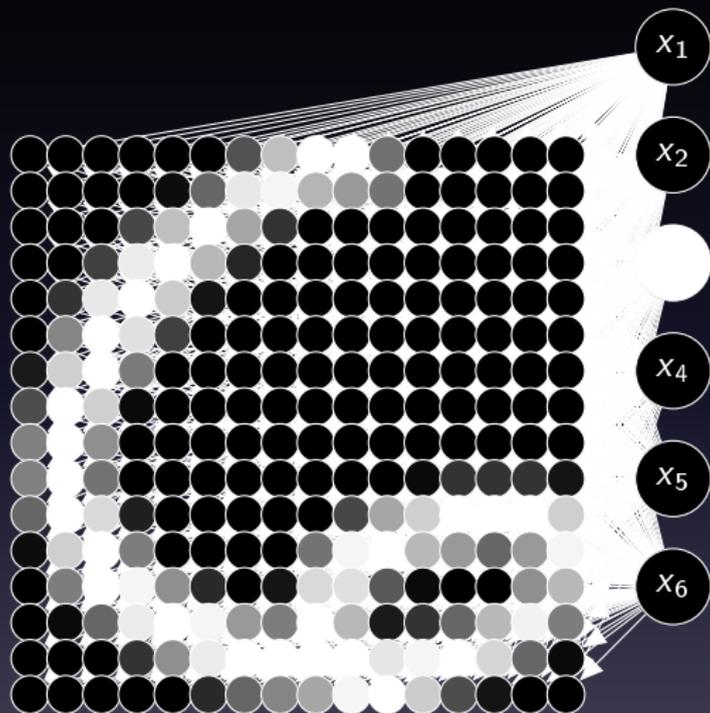
Latent Variable Model

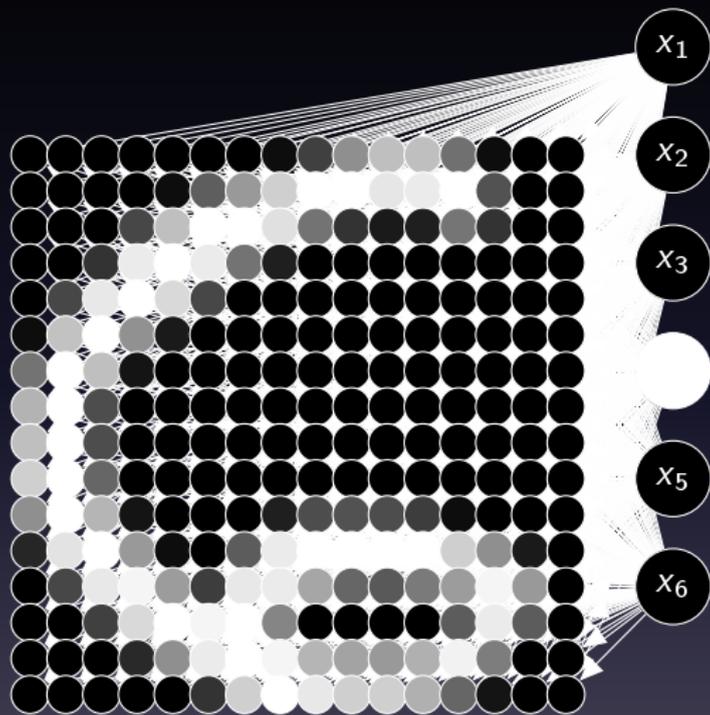


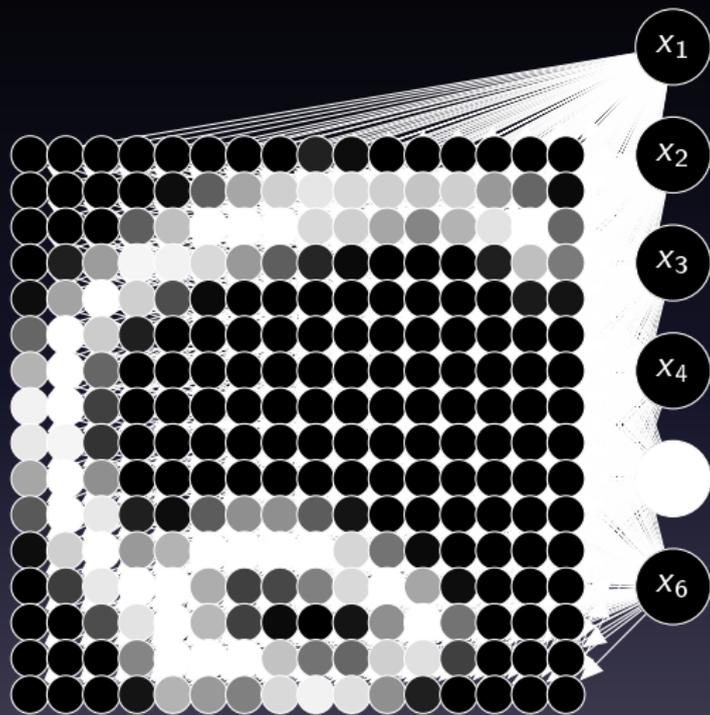


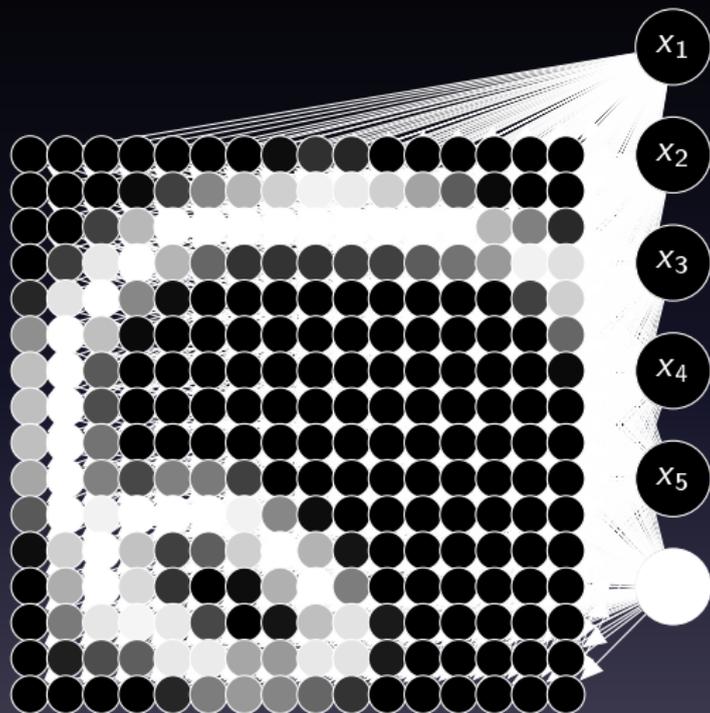












Template Matching

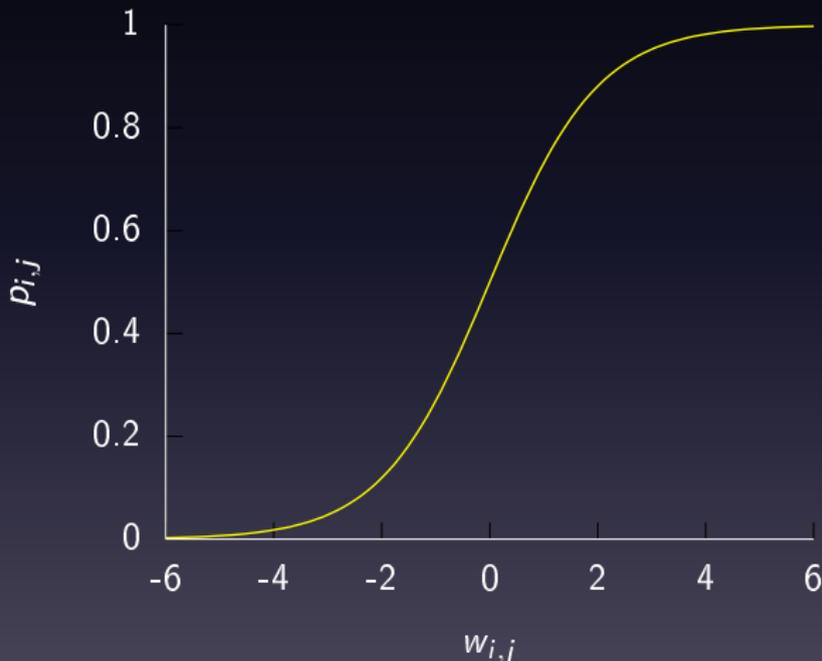
- Each latent node associated with a 'template' digit.
- If as many nodes as data then model is like 'nearest neighbour' with a particular distance measure.
- If less nodes than data then model is like a mixture of Bernoulli distributions.
- What if we allow several nodes to be switched on together?

Templates to Features

- In template matching i th 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 i th node is on the i th vector of probabilities is used.
- What if the i th node and the k th node are on?
 - How do we combine \mathbf{p}_i and \mathbf{p}_k to give probabilities of pixels?

Squashing Function

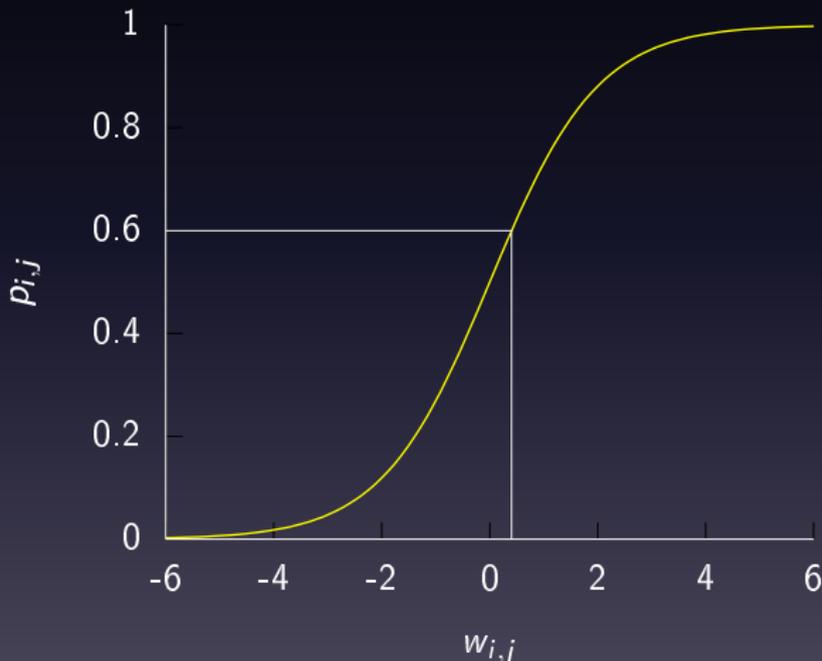
- One solution is to first reparameterise $p_{i,j}$ as a squashing function,



- For example the sigmoid function.

Squashing Function

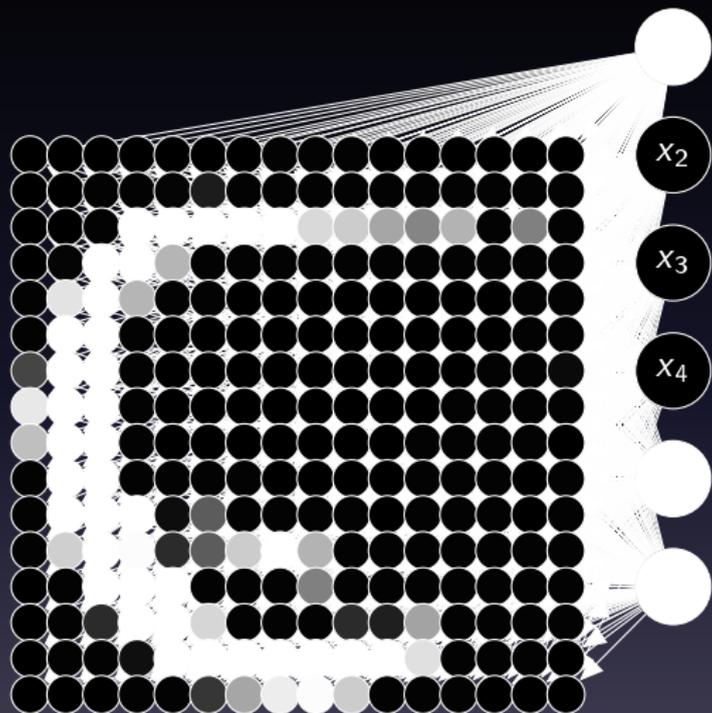
- One solution is to first reparameterise $p_{i,j}$ as a squashing function,

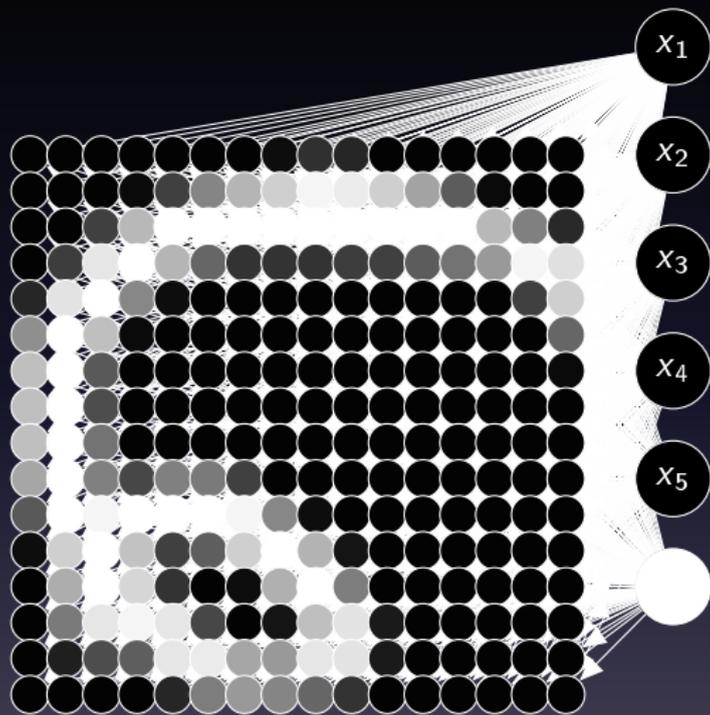


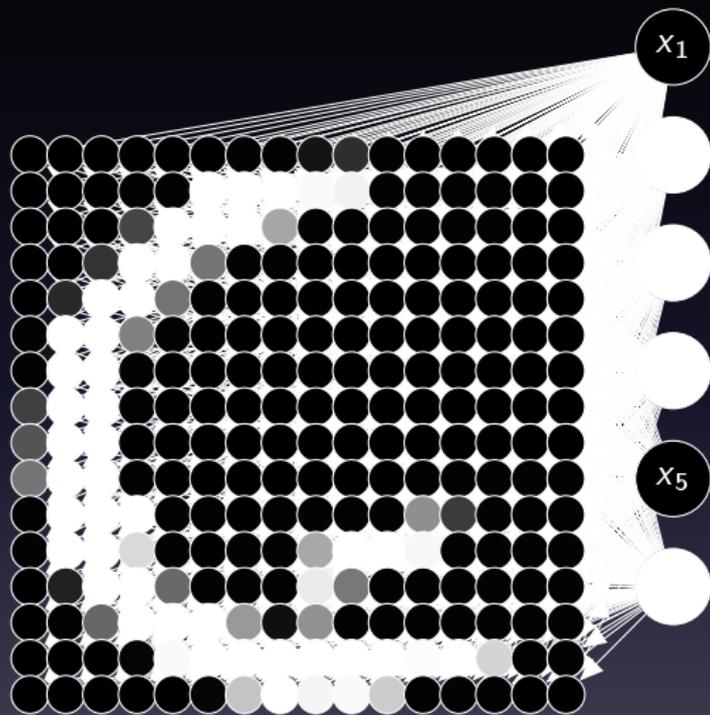
- For example the sigmoid function.

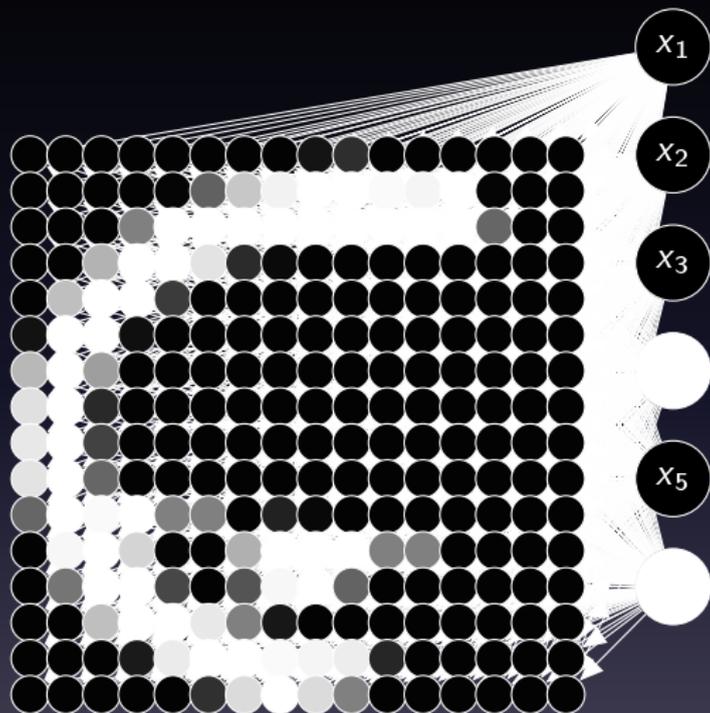
Addition Before Squashing

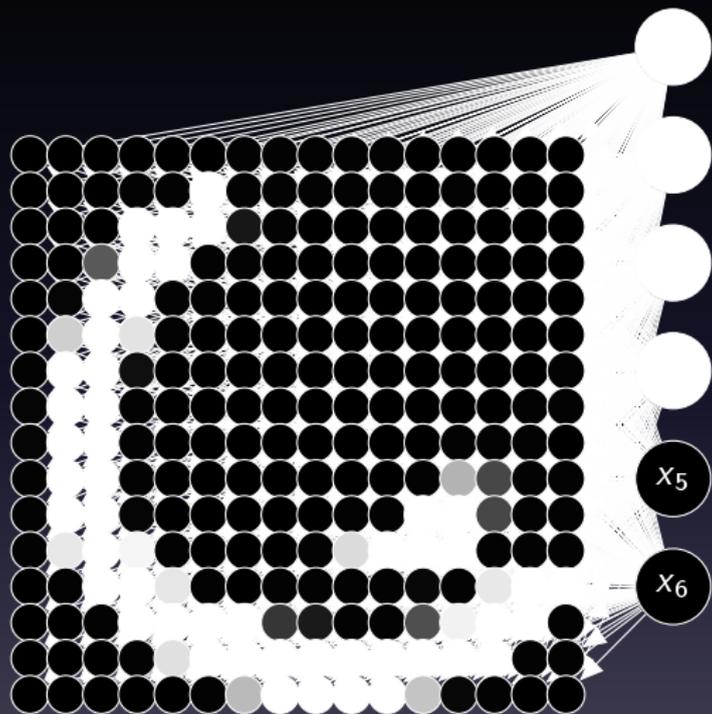
- Example: if latent node 1 and 6 are on.
- Can't add $p_{:,1}$ to $p_{:,6}$ to obtain probability that node is on.
- Instead add $w_{:,1}$ to $w_{:,6}$ and push through squashing function.
 - In general for $\mathbf{p}_{i,:}$ compute $\mathbf{W}\mathbf{x}_{i,:}$.
 - Then $p_{i,j} = \sigma(\mathbf{w}_{j,:}^\top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

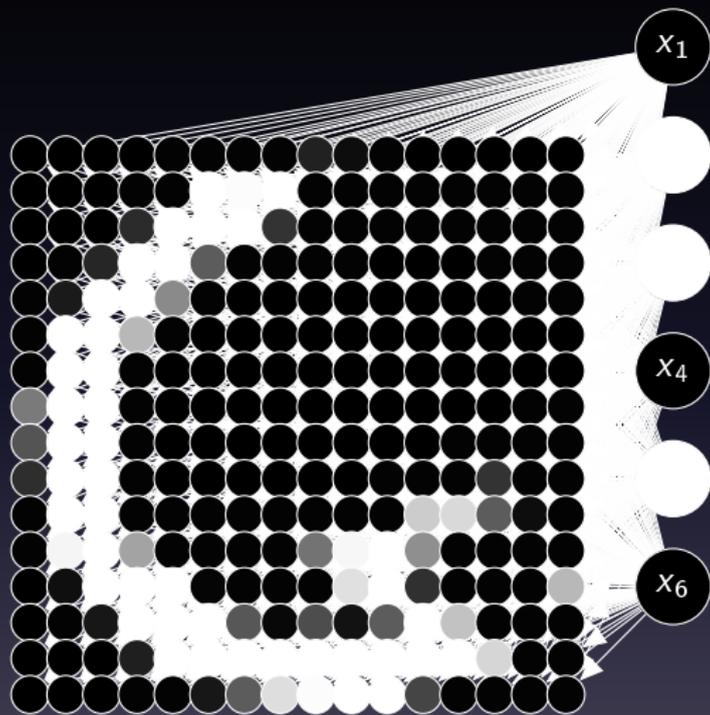






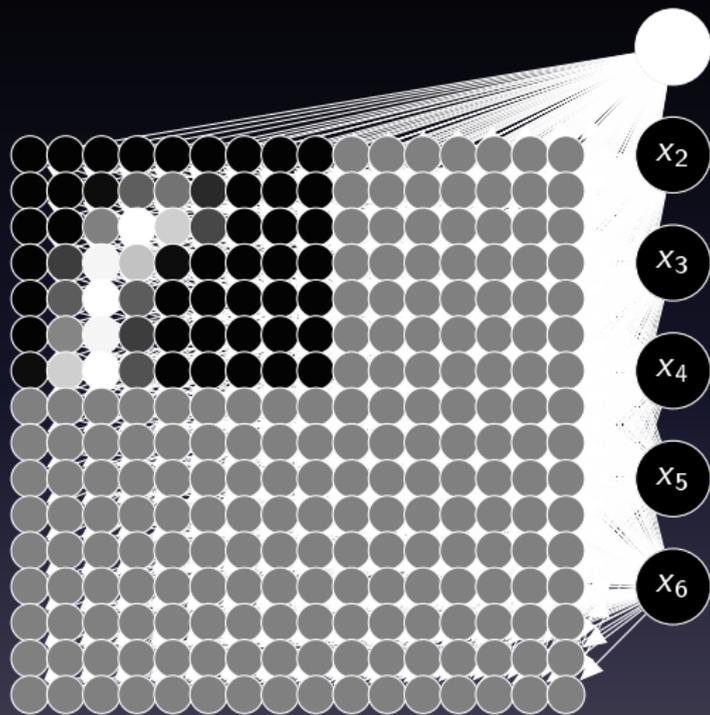


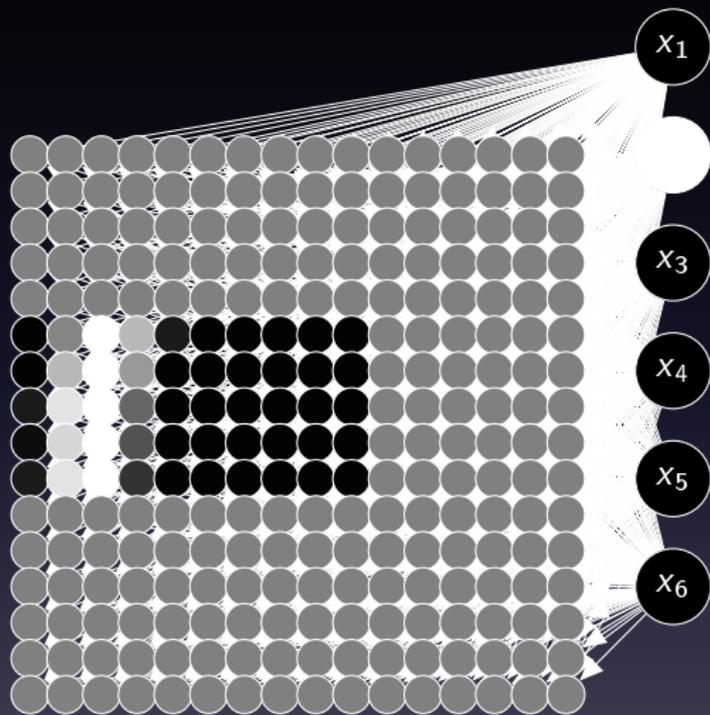


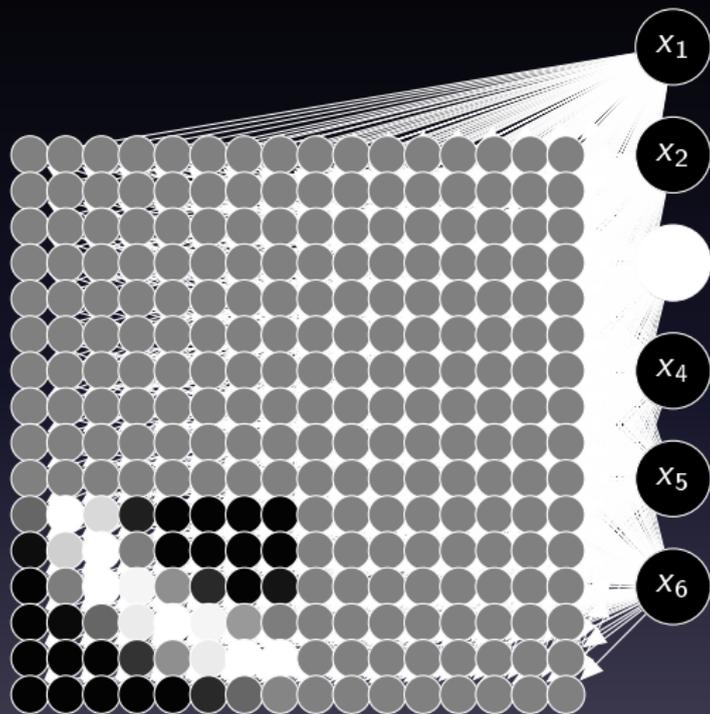


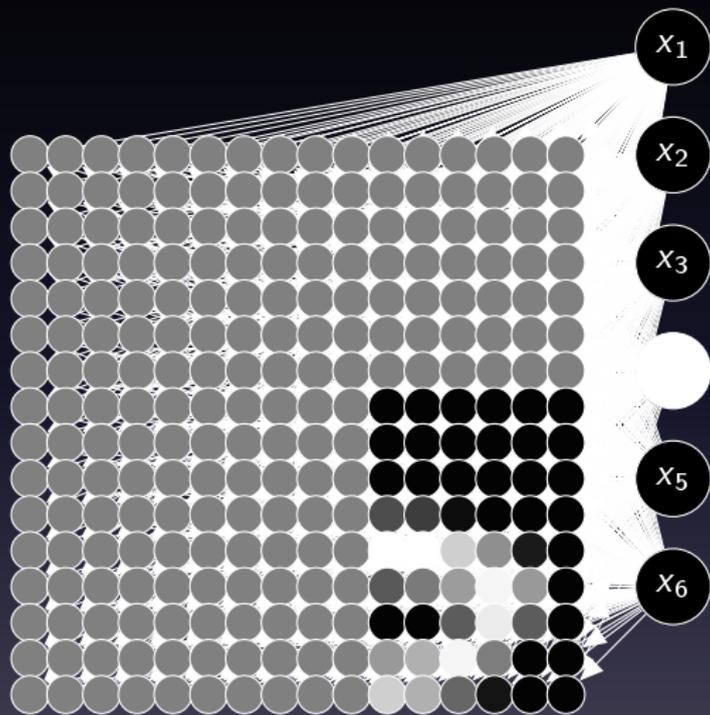
Localized Receptive Fields

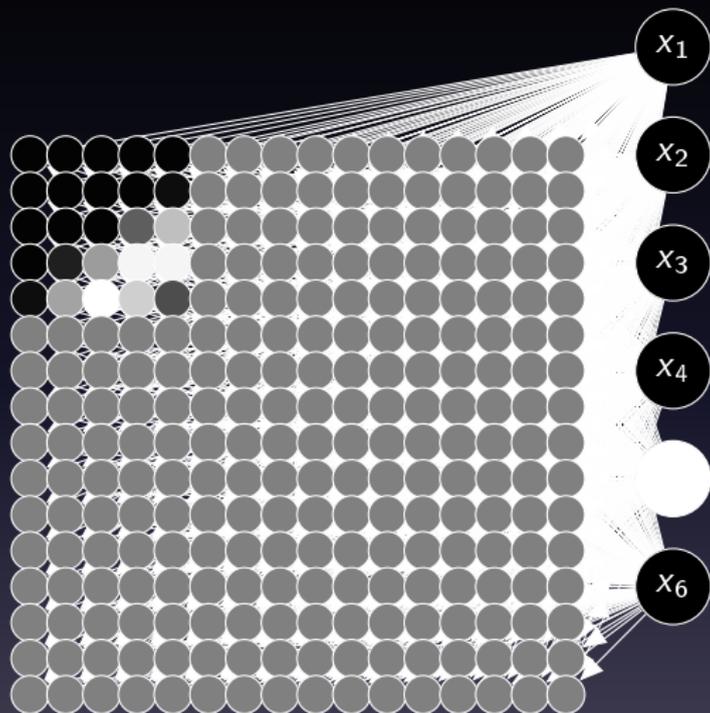
- Model can now fit global model as sum of parts.
- Each latent node associated with local features.
- Structure of model combines local features in products of experts manner (?).

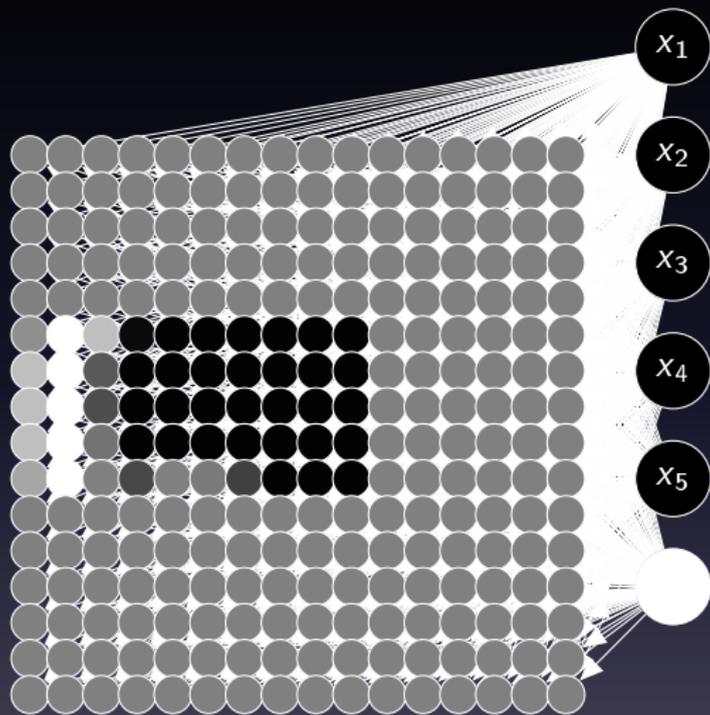


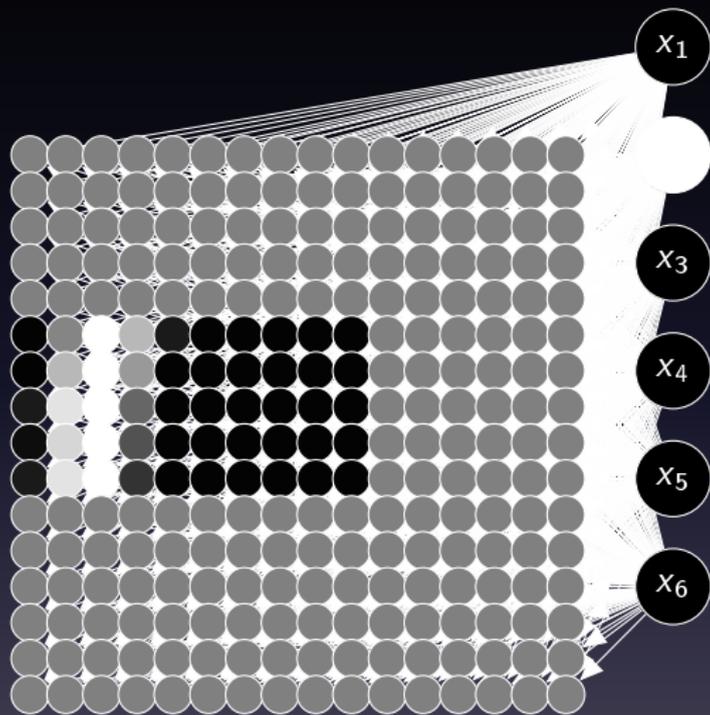


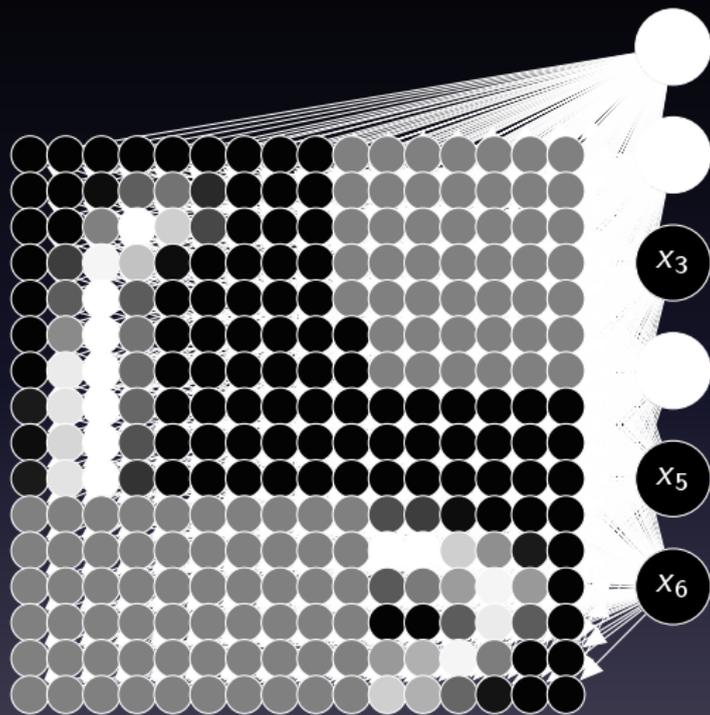


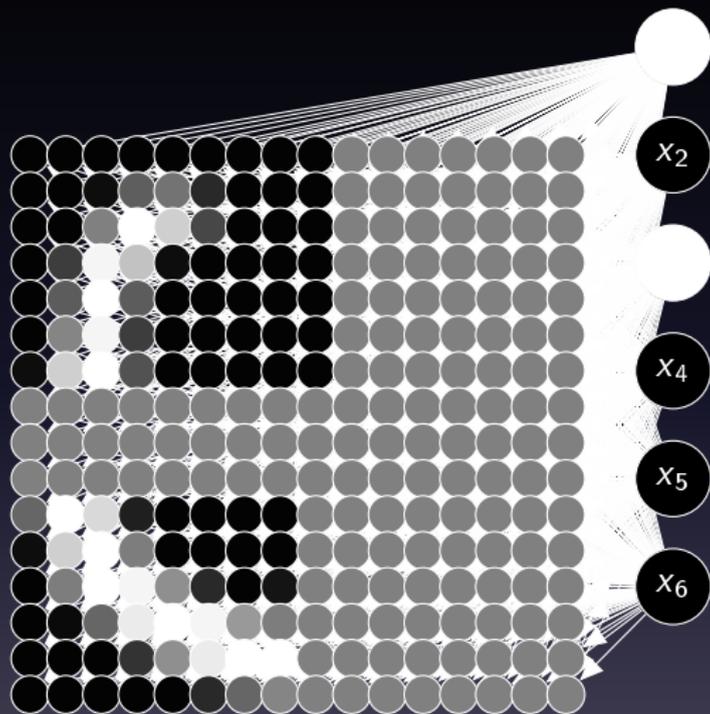


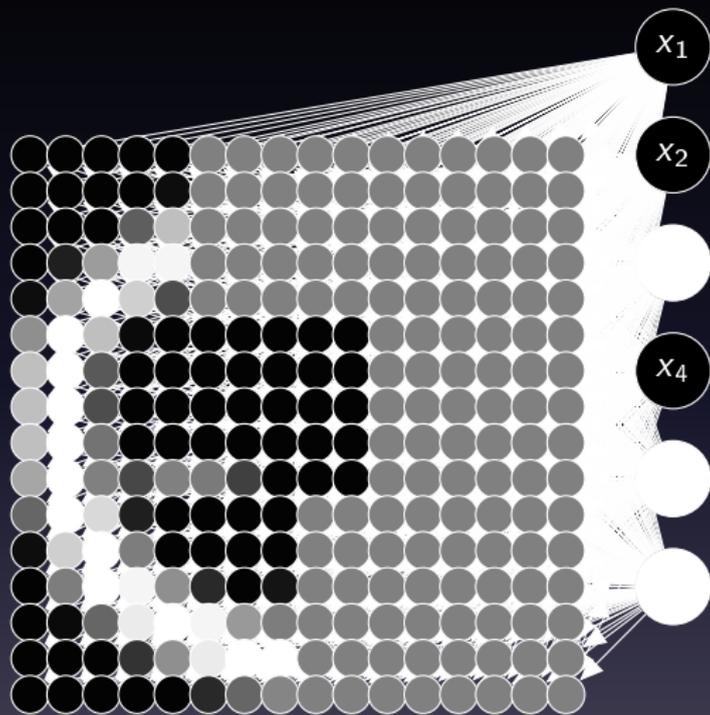


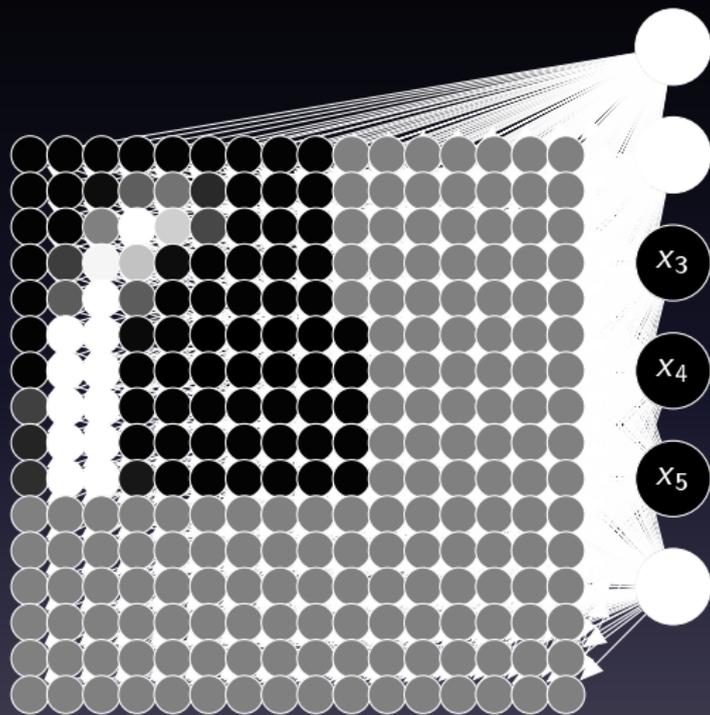


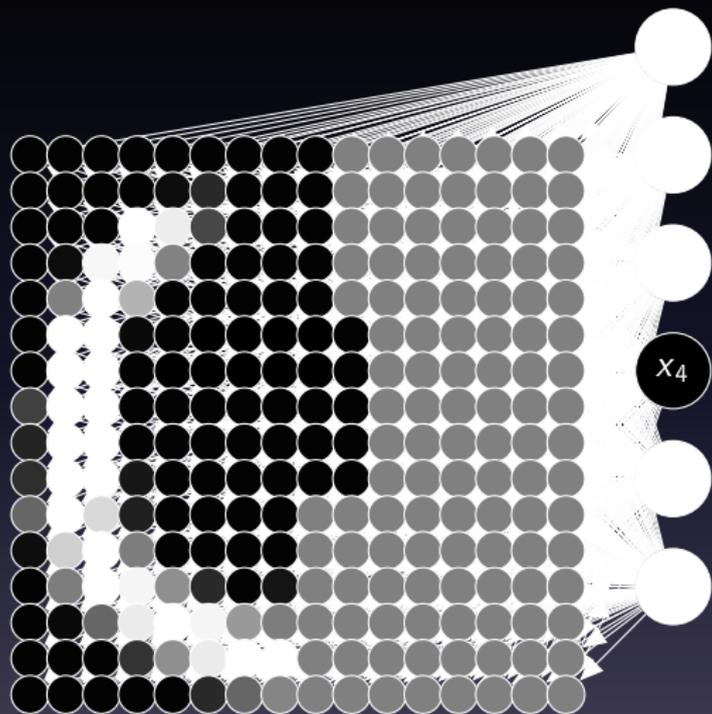












Restricted Boltzmann Machine

- 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 $\pi_j = \sigma(b_j)$ where $\sigma(\cdot)$ is the sigmoid function¹ and b_j is a 'bias' parameter.

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

Restricted Boltzmann Machine: Binomial Prior

- Parameterizing in this way means

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

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

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

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

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

which, because \mathbf{Y} is binary can be rewritten² as

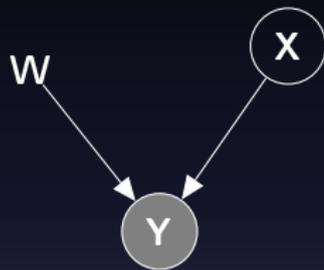
$$P(\mathbf{Y}|\mathbf{X}, \mathbf{W}) \propto \prod_{i=1}^n \exp\left(-(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})^{\top} \text{diag}(\mathbf{c}) (\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})\right)$$

²To complete the square where we extracted $\mathbf{x}_{i,:}^{\top} \mathbf{W}^{\top} \text{diag}(\mathbf{c}) \mathbf{W} \mathbf{x}_{i,:}$ 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* ... ??

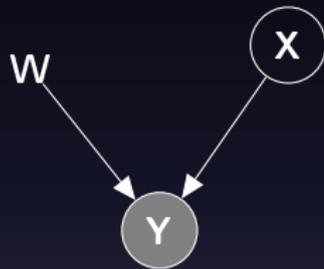


$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \prod_{j=1}^p p_{i,j}^{y_{i,j}} (1-p_{i,j})^{(1-y_{i,j})}$$

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* ... ??

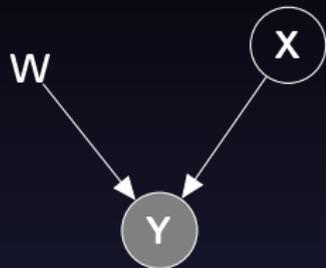


$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \prod_{j=1}^p p_{i,j}^{y_{i,j}} (1-p_{i,j})^{(1-y_{i,j})}$$

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* ... ??



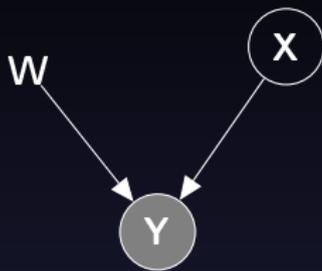
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \prod_{j=1}^p p_{i,j}^{y_{i,j}} (1-p_{i,j})^{(1-y_{i,j})}$$

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

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* ... ??

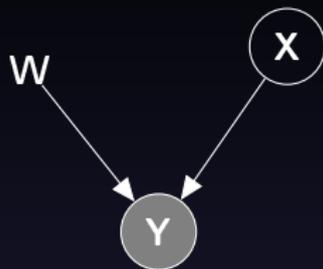


$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \prod_{j=1}^p p_{i,j}^{y_{i,j}} (1-p_{i,j})^{(1-y_{i,j})}$$

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

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

Marginalization of \mathbf{X}

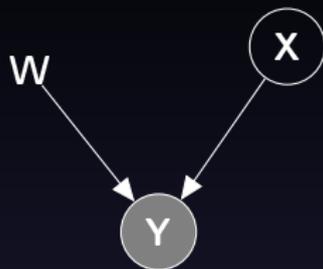


$$P(\mathbf{Y}|\mathbf{X}, \mathbf{W}) \propto \prod_{i=1}^n \exp\left(-(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})^\top \text{diag}(\mathbf{c}) (\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})\right)$$

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

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

Marginalization of \mathbf{X}

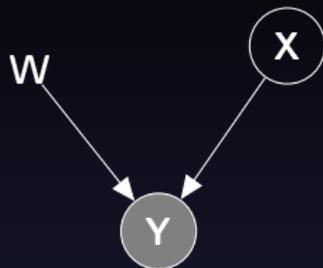


$$P(\mathbf{Y}|\mathbf{X}, \mathbf{W}) \propto \prod_{i=1}^n \exp\left(-(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})^\top \text{diag}(\mathbf{c}) (\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})\right)$$

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

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

Model Factorizes Across Data

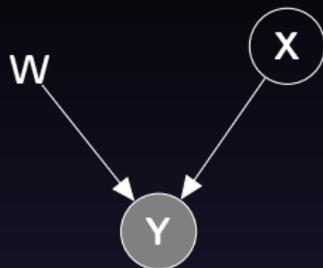


$$P(\mathbf{y}_{i,:} | \mathbf{x}_{i,:}, \mathbf{W}) \propto \exp\left(-(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})^\top \text{diag}(\mathbf{c})(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})\right)$$

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

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

Model Factorizes Across Data



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

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

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

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

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

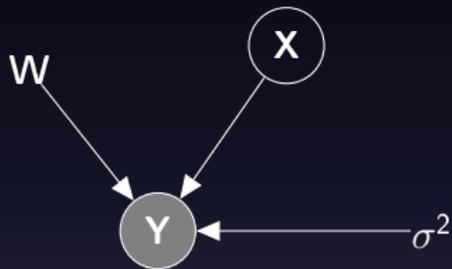
where

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

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*.

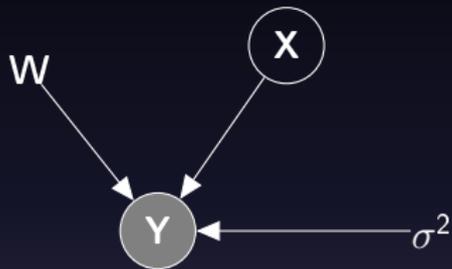


$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \mathcal{N}(y_{i,:} | \mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I})$$

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*.

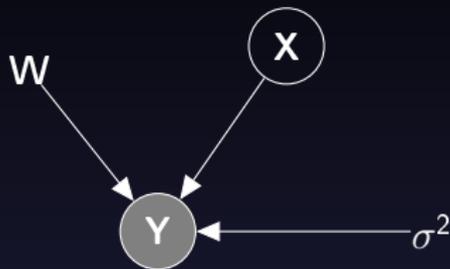


$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \mathcal{N}(\mathbf{y}_{i,:} | \mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I})$$

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*.



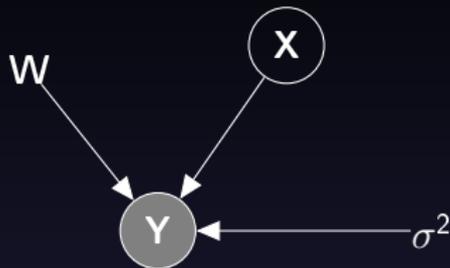
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \mathcal{N}(\mathbf{y}_{i,:} | \mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I})$$

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

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*.



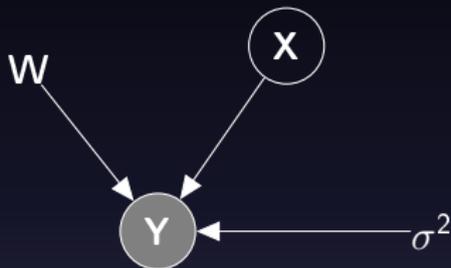
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^n \mathcal{N}(\mathbf{y}_{i,:} | \mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I})$$

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

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

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (?)



$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^n \mathcal{N}(y_{i,:} | \mathbf{0}, \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I})$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (?)

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

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

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

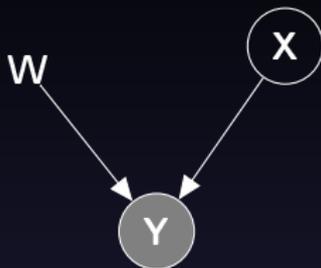
$$\mathbf{W} = \mathbf{U}_q \mathbf{L} \mathbf{R}^\top, \quad \mathbf{L} = (\boldsymbol{\Lambda}_q - \sigma^2 \mathbf{I})^{\frac{1}{2}}$$

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

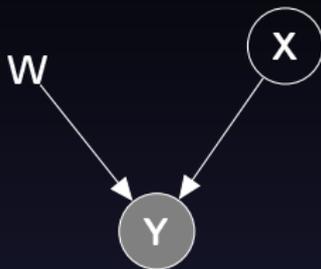


$$P(\mathbf{y}_{i,:} | \mathbf{x}_{i,:}, \mathbf{W}) \propto \exp\left(-(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})^\top \text{diag}(\mathbf{c})(\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})\right)$$

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

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

PCA and RBM

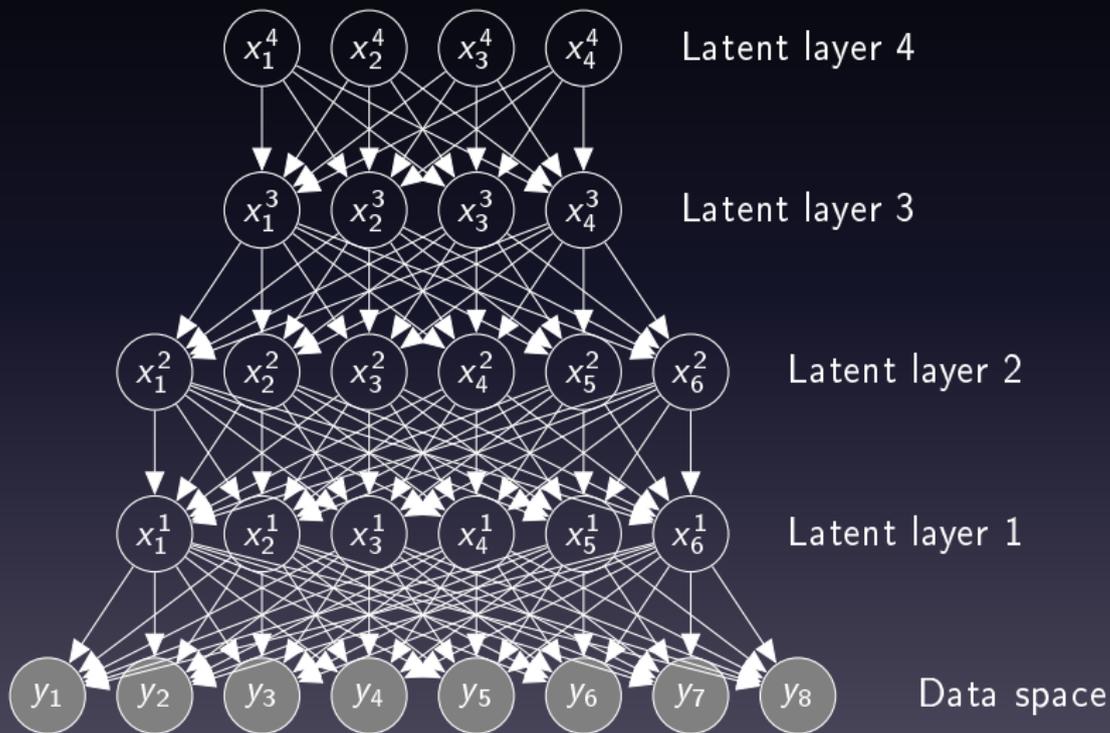


$$p(\mathbf{y}_{i,:} | \mathbf{x}_{i,:}, \mathbf{W}) \propto \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})^\top (\mathbf{y}_{i,:} - \mathbf{W}\mathbf{x}_{i,:})\right)$$

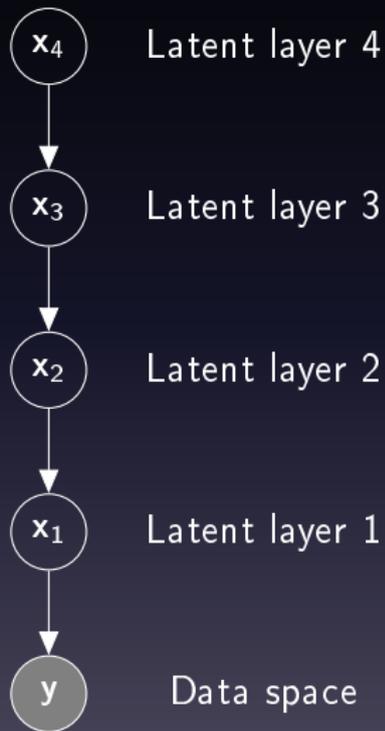
$$p(\mathbf{x}_{i,:}) \propto \exp\left(-\frac{1}{2} \mathbf{x}_{i,:}^\top \mathbf{x}_{i,:}\right)$$

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

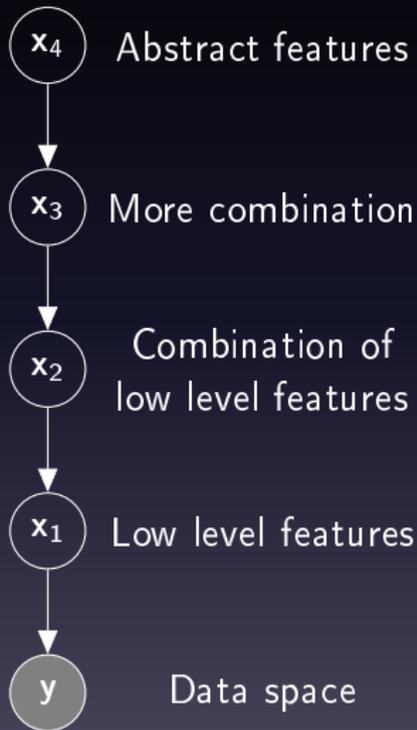
Deep Models



Deep Models



Deep Models

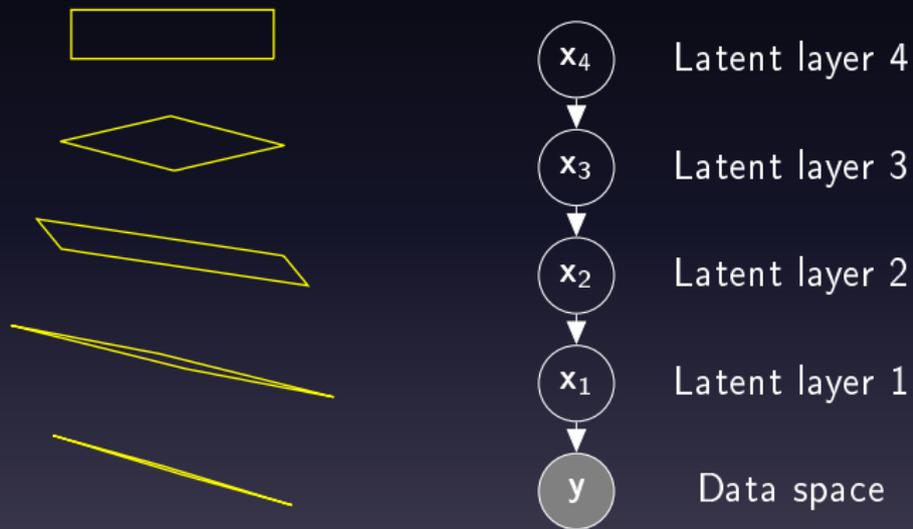


Deep Gaussian Processes

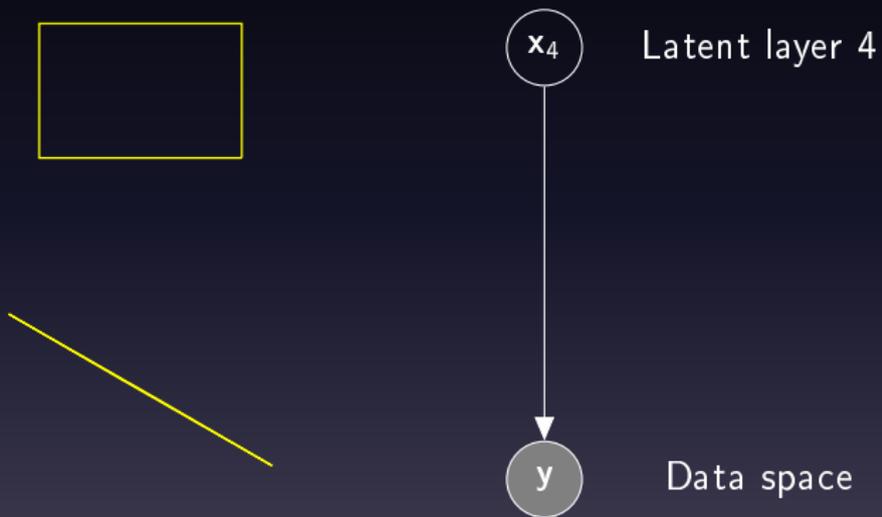
(?)

- Deep architectures allow abstraction of features (???)
- We use variational approach to stack GP models.

Stacked PCA



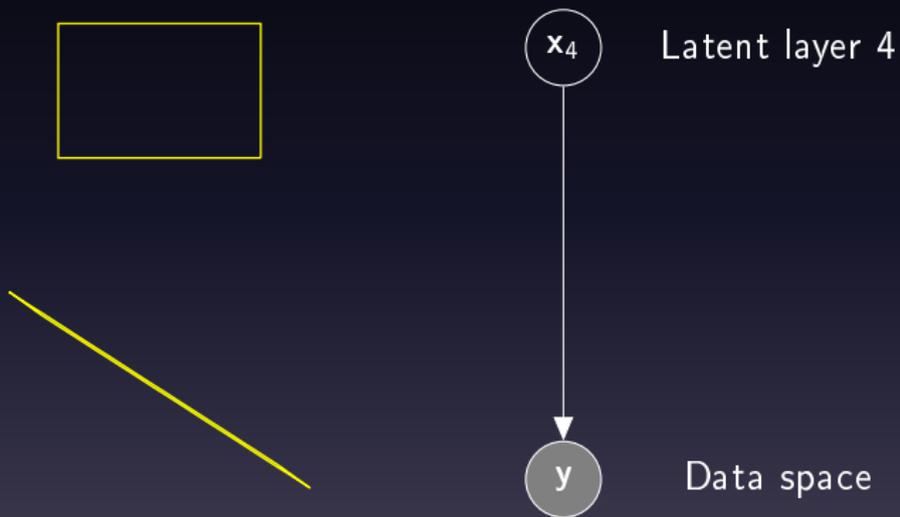
Stacked PCA



Stacked PCA



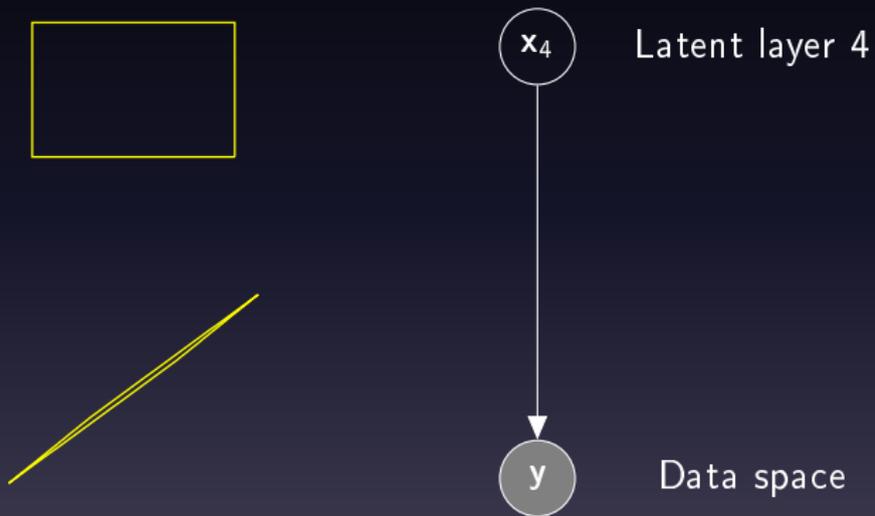
Stacked PCA



Stacked PCA



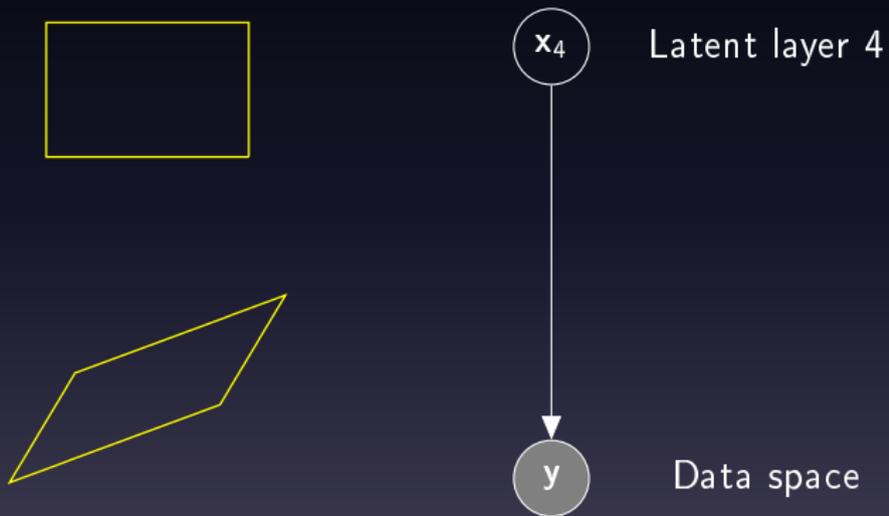
Stacked PCA



Stacked PCA



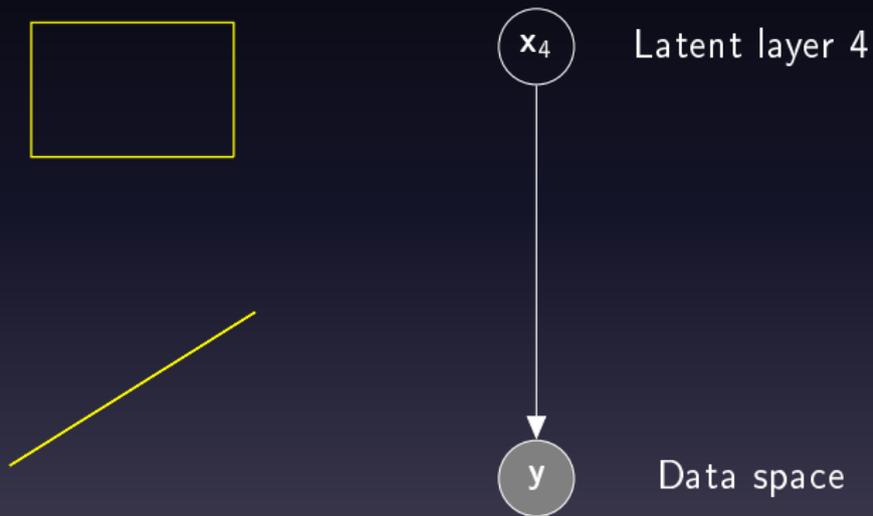
Stacked PCA



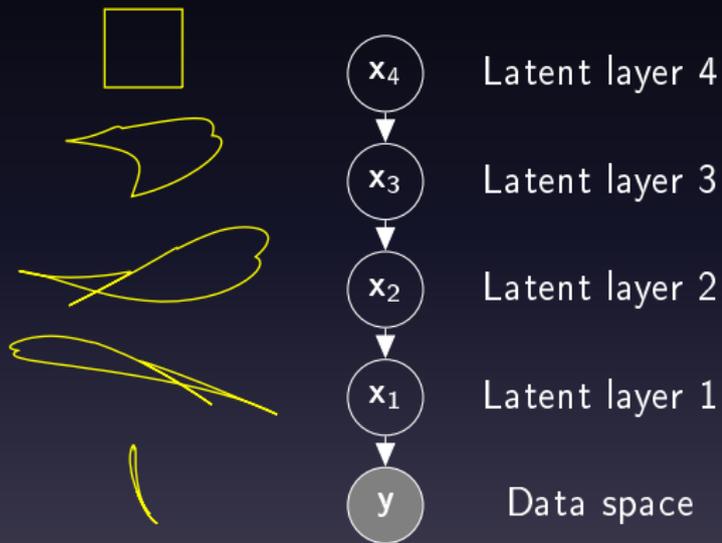
Stacked PCA



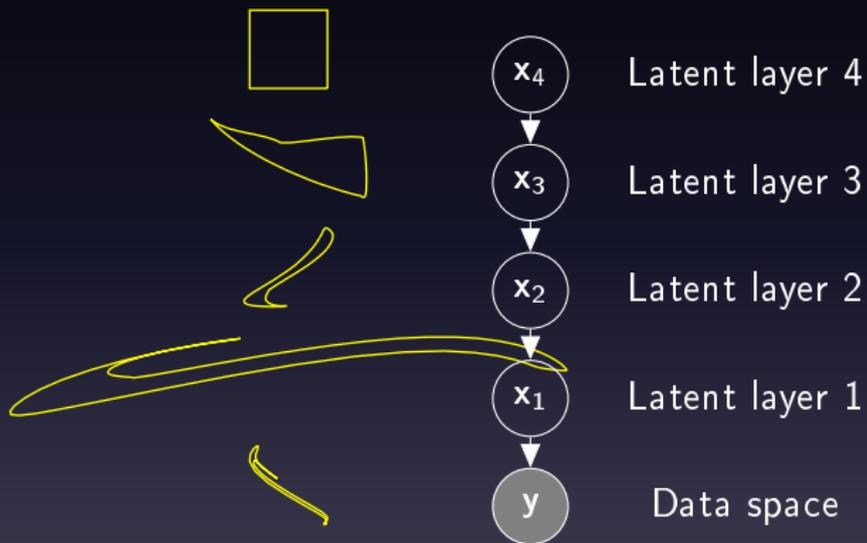
Stacked PCA



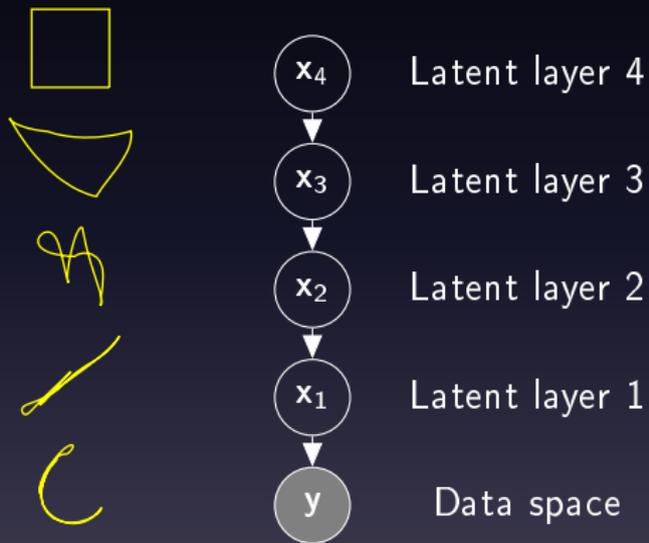
Stacked GPs



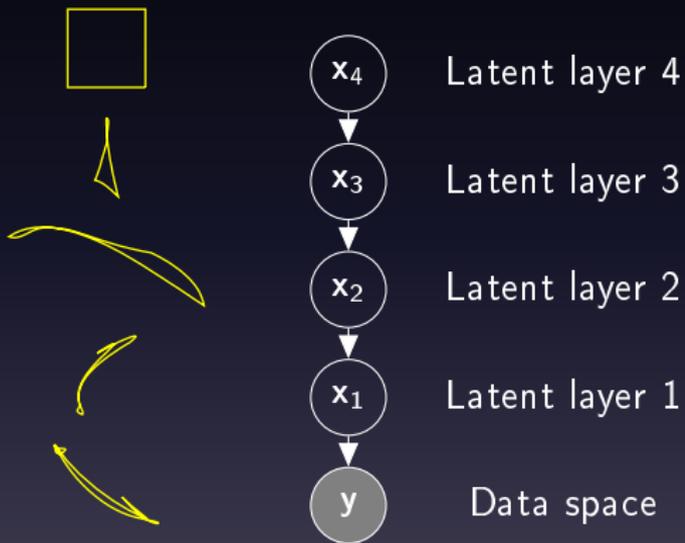
Stacked GPs



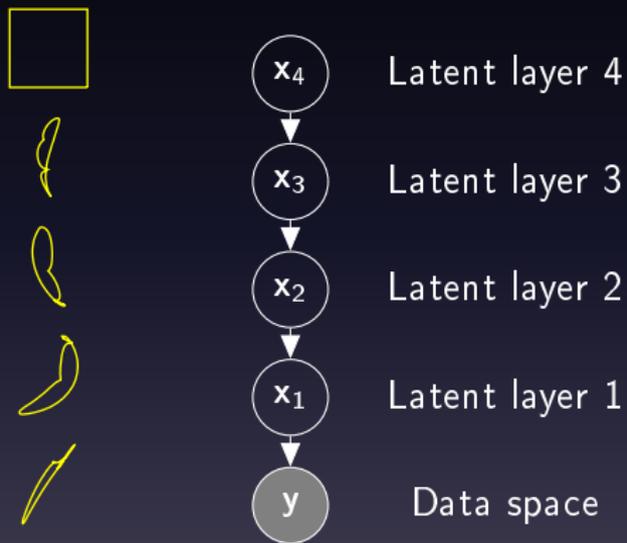
Stacked GPs



Stacked GPs



Stacked GPs



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

(?)

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

(?)

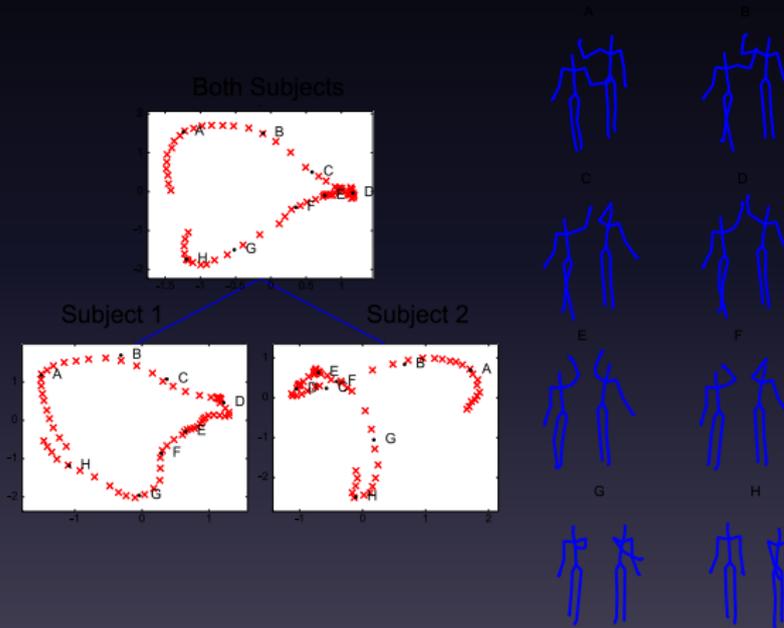


Figure: Hierarchical model of a 'high five'.

Within Subject Hierarchy

(?)

Decomposition of Body

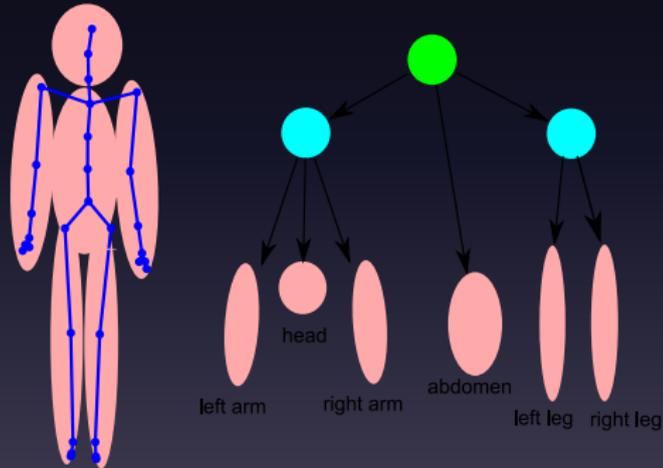


Figure: Decomposition of a subject.

Single Subject Run/Walk

(?)

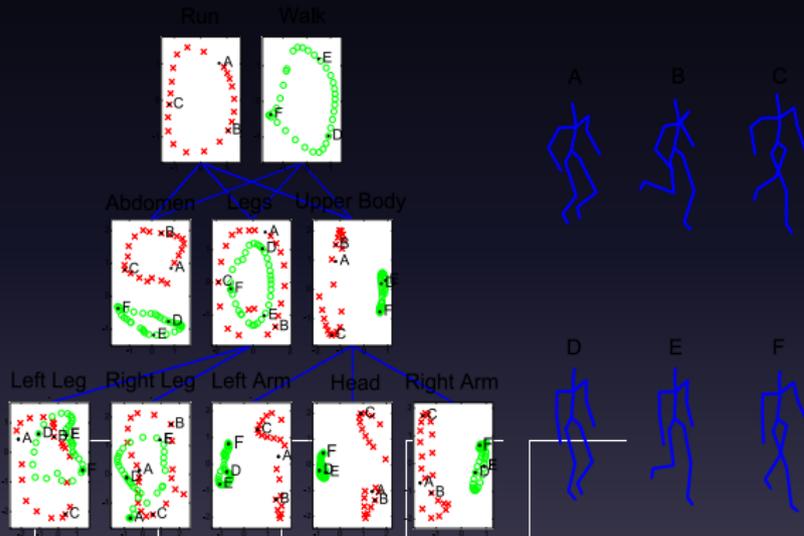
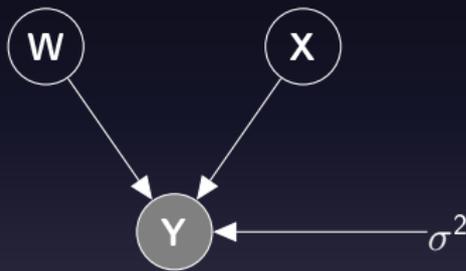


Figure: Hierarchical model of a walk and a run.

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. (??)
 - Requires expectations of $\mathbf{K}_{f,\mathbf{u}}$ and $\mathbf{K}_{f,\mathbf{u}}\mathbf{K}_{\mathbf{u},f}$.

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. ???).

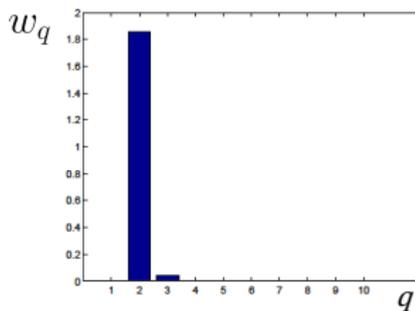
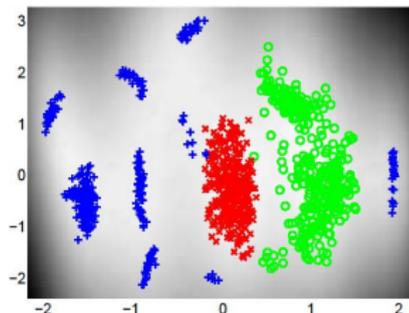
Automatic dimensionality detection

- 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

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

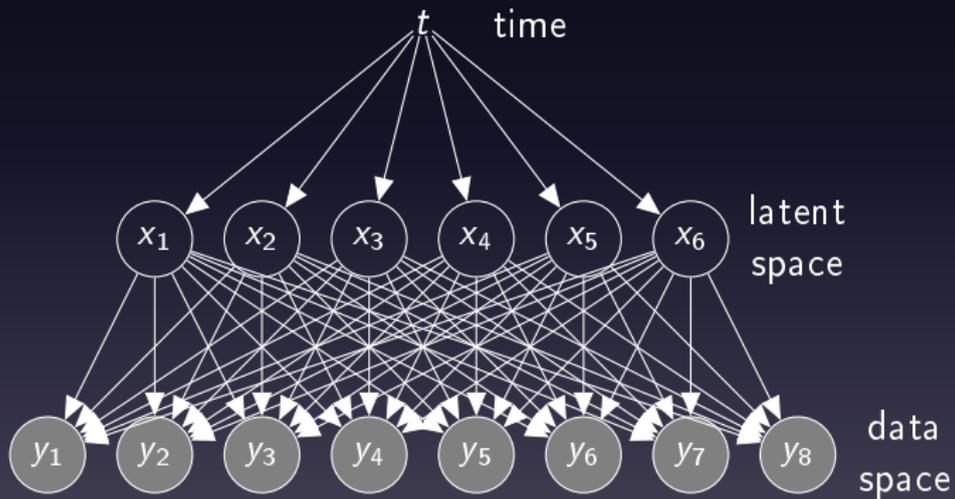
- Example



Face Demo

Gaussian Process Dynamical Systems

(?)



Gaussian Process over Latent Space

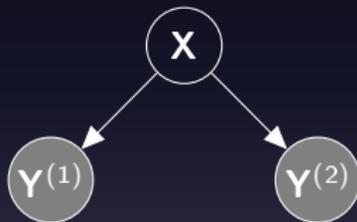
- Assume a GP prior for $p(\mathbf{X})$.
- Input to the process is time, $p(\mathbf{X}|t)$.

Gaussian Process over Latent Space

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

Modeling Multiple 'Views'

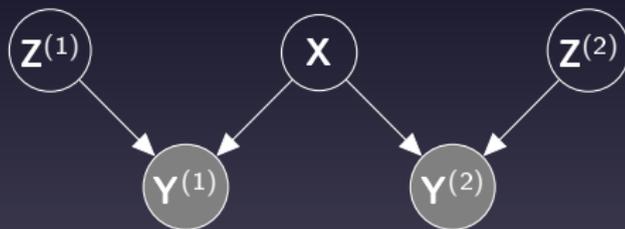
- Single space to model correlations between two different data sources, e.g., images & text, image & pose.
- Shared latent spaces: (???)



- 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

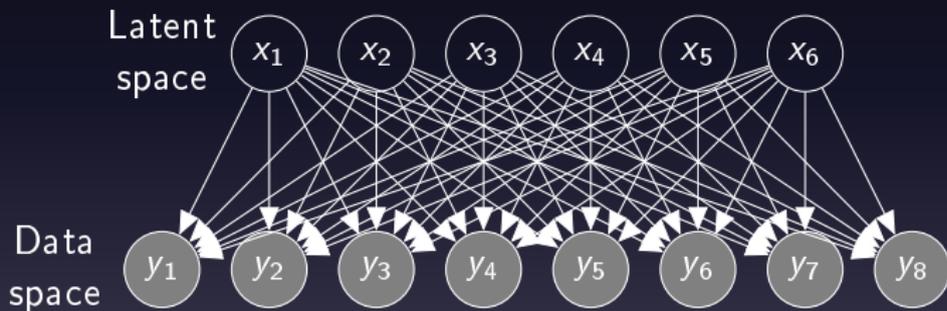
- In real scenarios, the 'views' are neither fully independent, nor fully correlated.
- Shared models
 - either allow information relevant to a single view to be mixed in the shared signal,
 - or are unable to model such private information.
- Solution: Model shared and private information (???????)



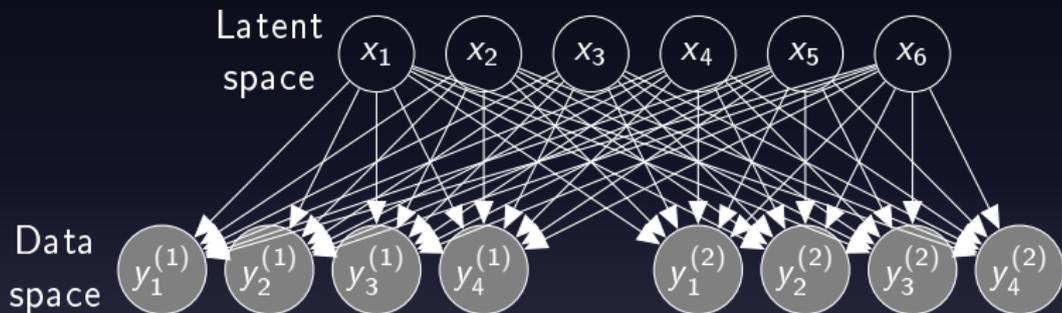
- Probabilistic CCA is case when dimensionality of \mathbf{Z} matches $\mathbf{Y}^{(i)}$ (cf Inter Battery Factor Analysis (?)).

Manifold Relevance Determination

(?)



Shared GP-LVM



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

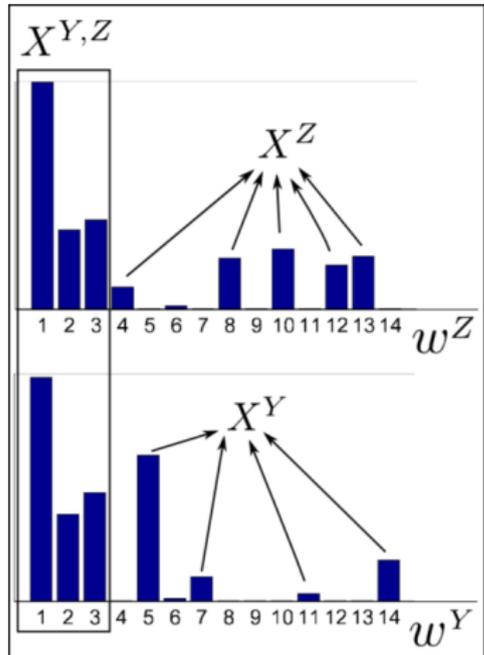
Example: Yale faces



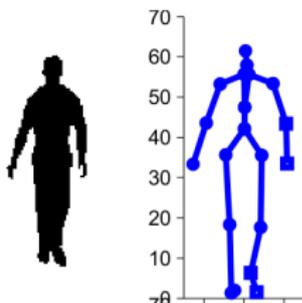
- Dataset Y: 3 persons under all illumination conditions
- Dataset Z: As above for 3 different persons
- Align datapoints \mathbf{x}_n and \mathbf{z}_n only based on the lighting direction

Results

- Latent space X initialised with 14 dimensions
- Weights define a segmentation of X
- Video / demo...



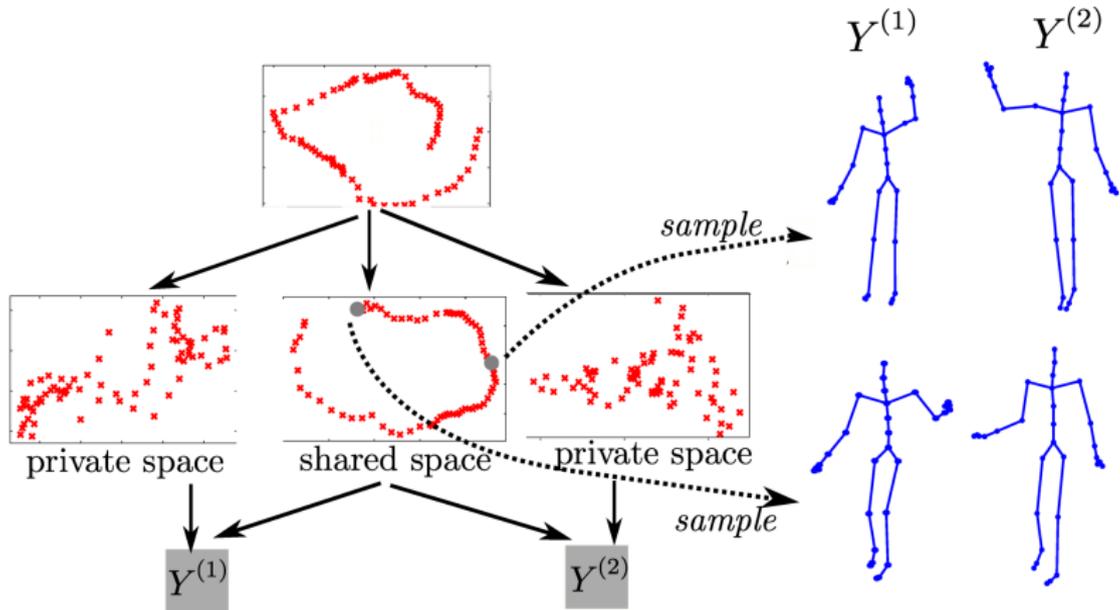
Potential applications..?



Motion Capture

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

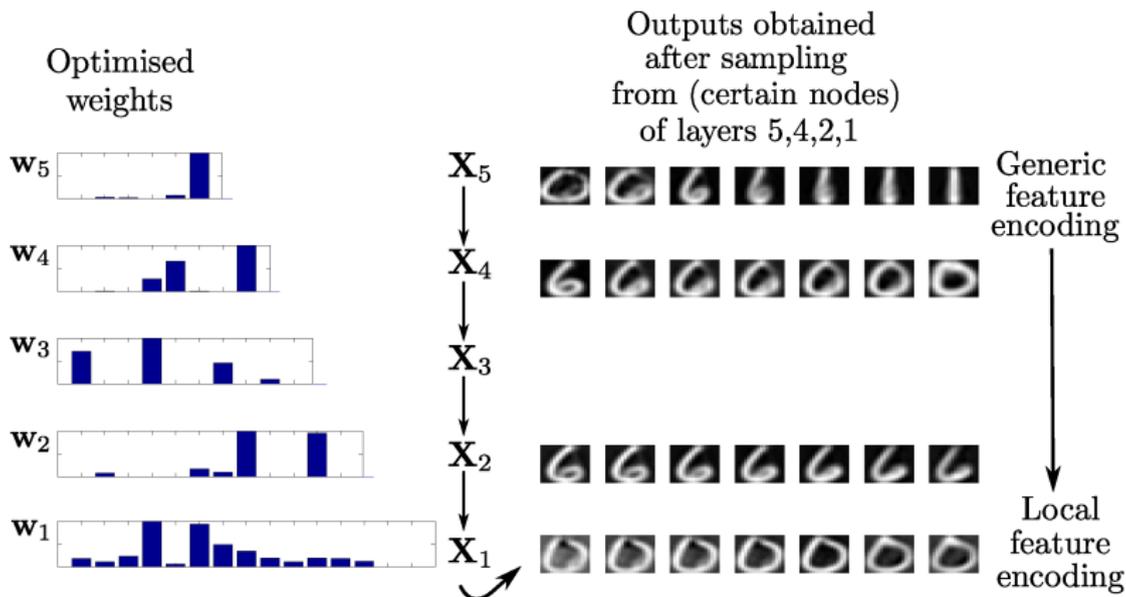
Deep hierarchies – motion capture



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](#)].
- A. Damianou, C. H. Ek, M. K. Titsias, and N. D. Lawrence. Manifold relevance determination. In ?. [[PDF](#)]. To appear.
- A. Damianou and N. D. Lawrence. Deep Gaussian processes. In C. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics*, volume 31, AZ, USA, 2013. JMLR W&CP 31. [[PDF](#)].
- A. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In P. Bartlett, F. Peirreira, C. Williams, and J. Lafferty, editors, *Advances in Neural Information Processing Systems*, volume 24, Cambridge, MA, 2011. MIT Press. [[PDF](#)].
- 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. Boullard, 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 ?. [[Google Books](#)].

References II

- A. Klami and S. Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72:39–46, 2008.
- J. Langford and J. Pineau, editors. *Proceedings of the International Conference in Machine Learning*, volume 29, San Francisco, CA, 2012. Morgan Kaufman. To appear.
- N. D. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. In ?, 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.
- 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 Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.
- 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](#)].

References III

- M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In Y. W. Teh and D. M. Titterton, 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.
- A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In ?. To appear.