Unsupervised Learning

Oliver Schulte

School of Computing Science

Simon Fraser University

Introduction to Deep Learning

Overview

- Unsupervised Learning: No label is given
- Tasks:
  - Generate Data
  - Component Analysis
    - Learning Latent Features (Basis Functions)
    - Reduce Dimensionality: merge correlated columns
- Methods
  - Auto-Encoder Neural Networks
    - Principal component analysis
  - Convolutional Auto-Encoder
  - Variational Auto-Encoder

Unsupervised Learning and Embeddings

- In many settings we are not given labels for observed data
  - E.g. text and videos on-line
- It is still possible and useful to learn embeddings = latent features
  - Dimensionality reduction
  - Eliminate redundancies among features.
  - Can help with missing data.
  - Generate instances
    - E.g. generate text, generate images
  - Support downstream supervised learning

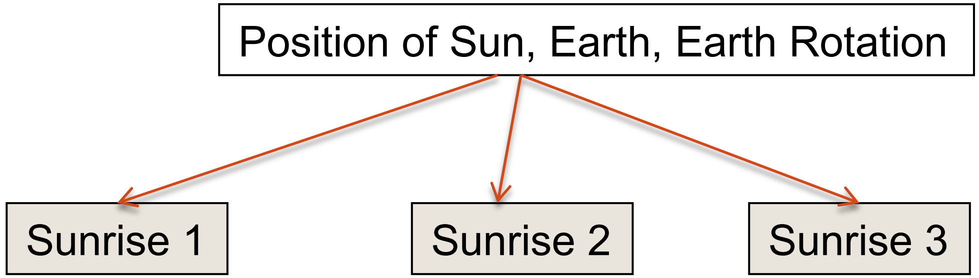
Component Analysis

Intuition

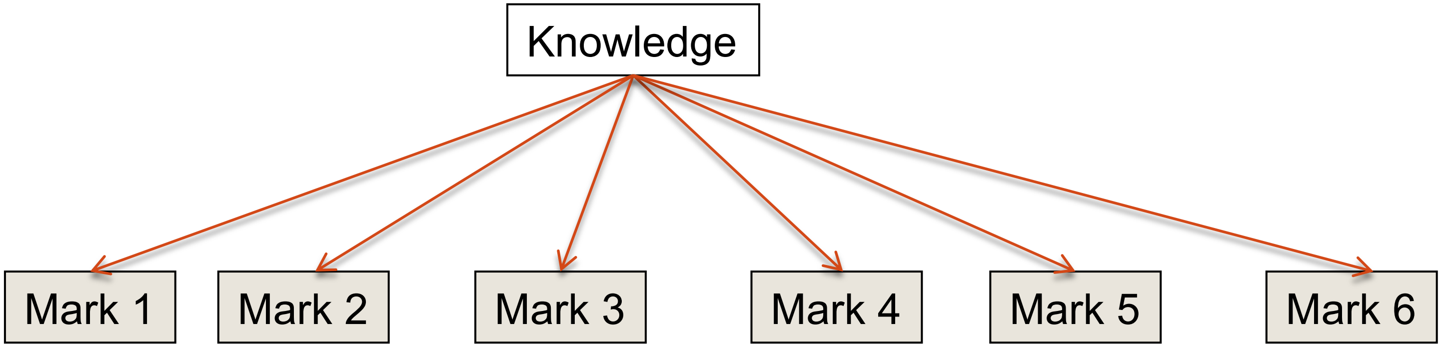
- Many high-dimensional data are generated from a few **unobserved** settings
- Example: *Digit Rotation*
  - Take a single 3 in a 28x28 = 784 pixel image.
  - Make new 3s by
    1. Rotating through an angle c1
    2. Shifting up or down c2
  - Every new 3 can be represented by two numbers (c1,c2).
  - Much less than 784!

Intuitive Examples

- Sun rose at 6 am each day this week.



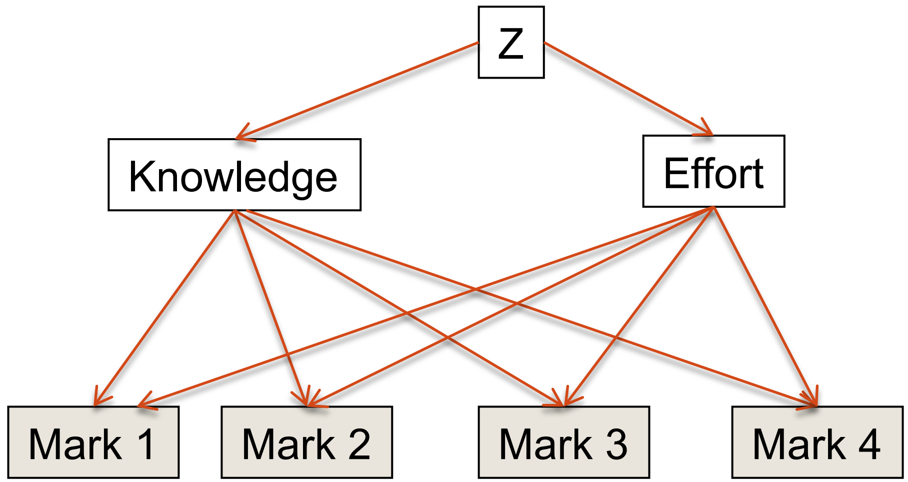
- Student's marks are explained by their knowledge.



Source Separation

- Source Separation Demo
  - In a noisy room, two microphones produce a combined signal.
  - Can you reconstruct the component signals from the combined signal?
  - Netflix ratings: Mum, Dad, Adriana and Marina all rate movies on one account. Can Netflix tell which ratings came from which person?

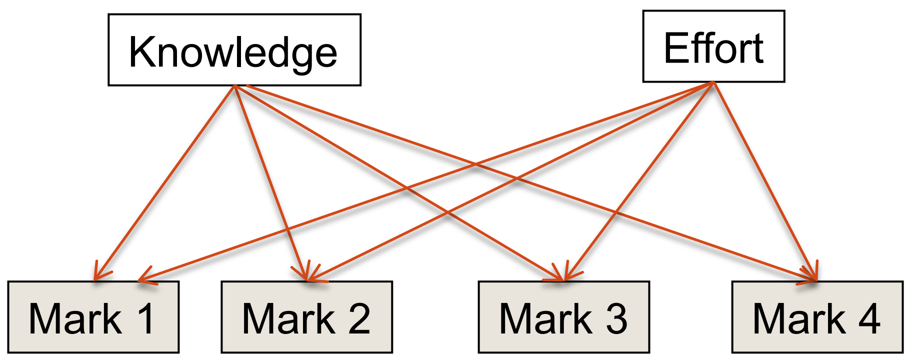
Hierarchical Version



- Latent variables at layer 1 explain correlations among observed variables.
- Latent variables at layer 2 explain correlations among latent variables at layer 1.
- This is the basic idea of a **deep belief network**.
  - Also convolutional neural network

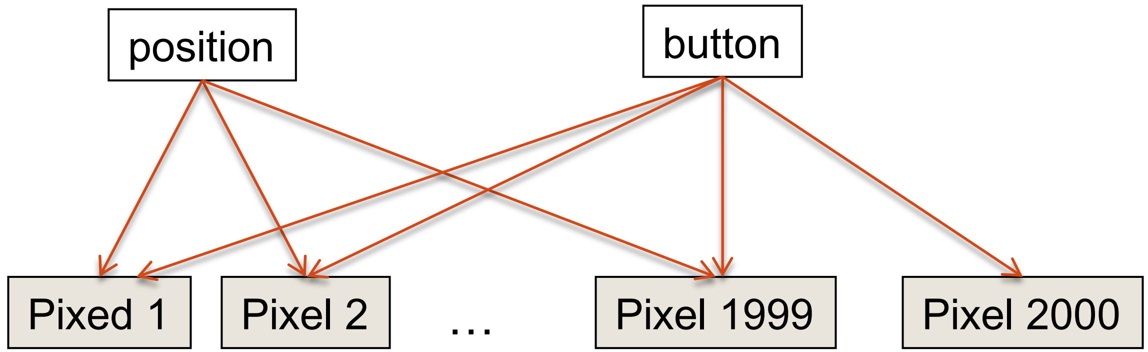
Latent Feature Dimensionality

- Assumption: The number of unobserved components m is much less than the number n of observed variables.
- The number of latent dimensions m can be specified by the user, or learned.

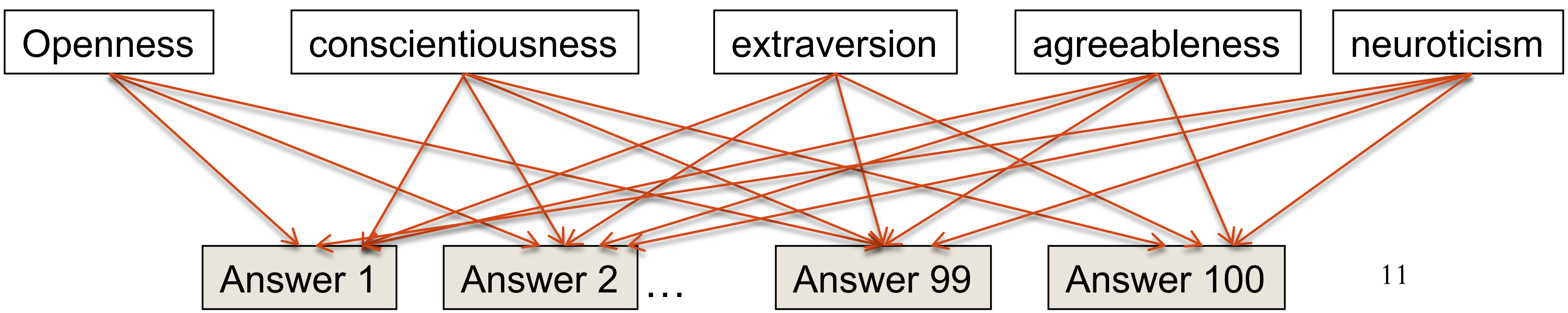


More Examples

- Video Game Screen: 4000 pixels (say) determined by two variables (joystick position, button pressed).



- The Big 5 in Psychology. Personality Types can be explained by 5 basic traits.



Consequence: Dimensionality Reduction

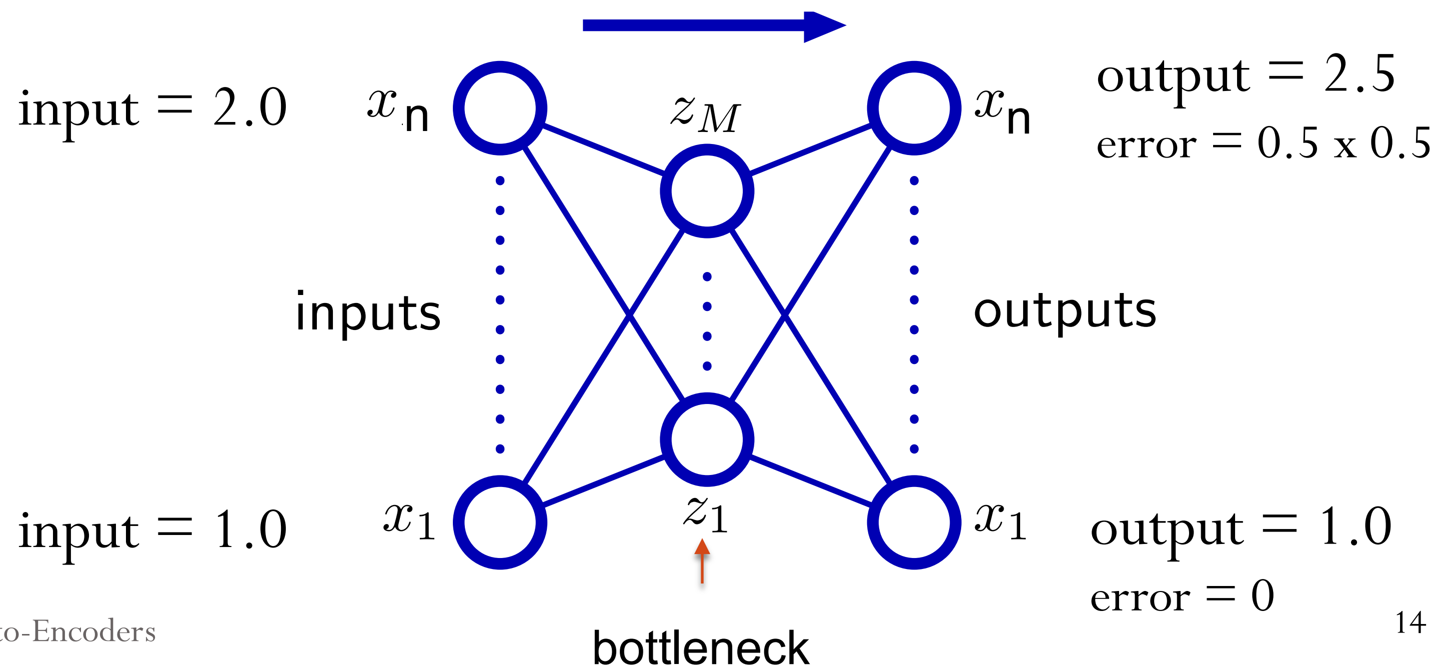
- If a tuple of n observed feature values is well predicted by m latent features, we can represent the n-tuple instead by a m-tuple, without (much) loss of information.
  - The data lie on a m-dimensional *manifold*.
  - Latent feature learning can be used to reduce the data dimensionality.
- Video Game example: instead of specifying all 2000 pixels, specify (position of joystick, button pressed).
  - From 2 numbers we can **reconstruct** 2000.

Learning Methods

Neural Net Auto-Encoders

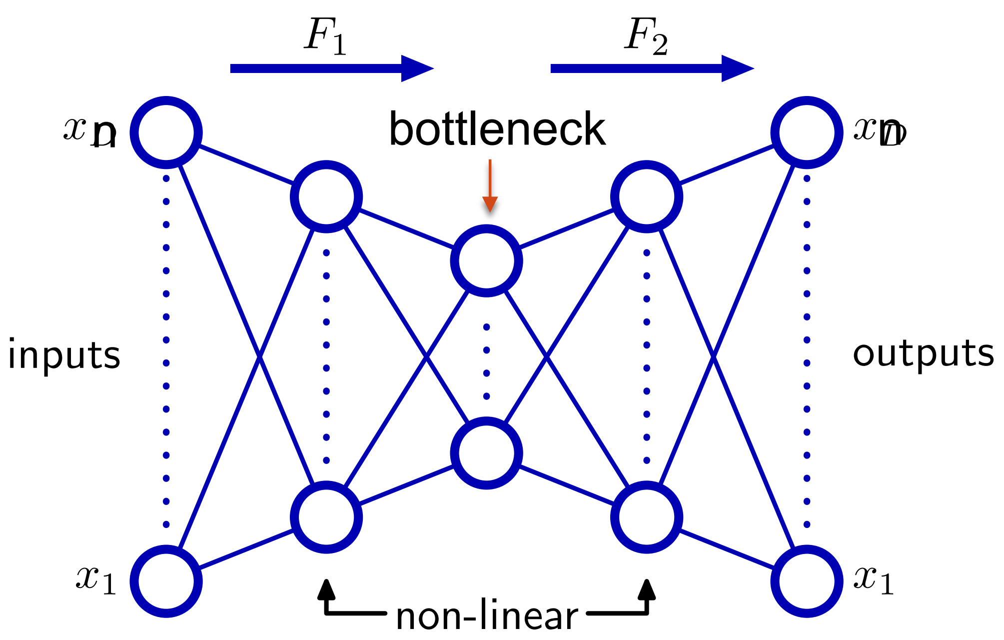
Auto-Association

- An **auto-associative** neural net has just as many input units as output units.
- The <u>error is the squared difference</u> between input unit x_i and output unit o_i.
- ➢ Backpropagation trains the network to recreate the input.
- The hidden layer maps n input values to m hidden node output values.
- If $n>m$ ➔ dimensionality reduction!



Deep Auto-Encoders

With more than one hidden layer, auto-encoders perform <u>non-linear dimensionality reduction</u>.

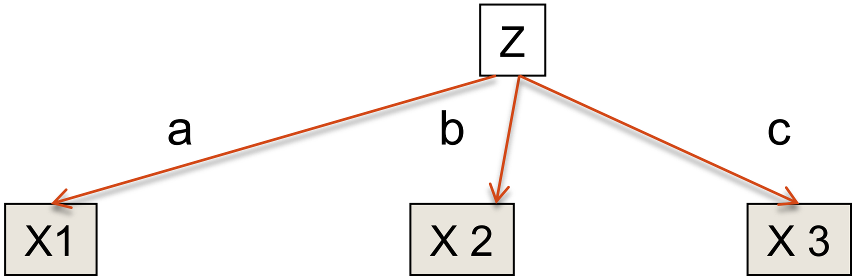


Learning Methods

Principal Component Analysis

Principal Component Assumption

- Observed Features are linear combinations of latent features.
  - Plus some noise.



$$X1 = a\,Z + \varepsilon$$

$$X2 = b\,Z + \varepsilon$$

$$X3 = c\,Z + \varepsilon$$

Principal Component Analysis

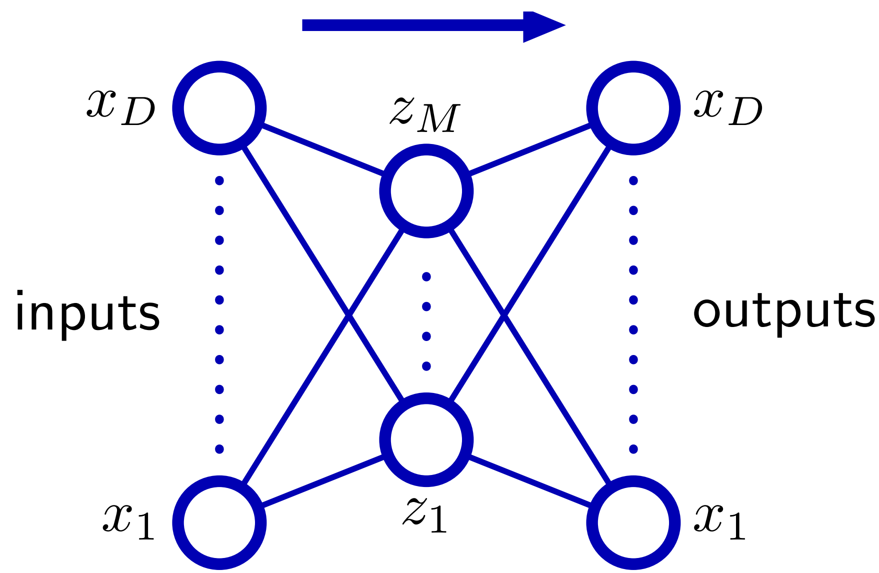
- Given number *m* of principal components, relatively easy to find optimal latent features.
- Columns are linear combinations of each other (neglecting noise), e.g.
  - X1 = a Z and X2 = b Z implies X2 = b/a X1.
- There is a closed-form solution for finding the set of optimal components with minimal reconstruction error.
  - Ranks components by importance (eigenvalues)
  - PCA demo

Eigenfaces

- http://en.wikipedia.org/wiki/Eigenface
- Regular human faces are reconstructed as linear combinations of the eigenfaces

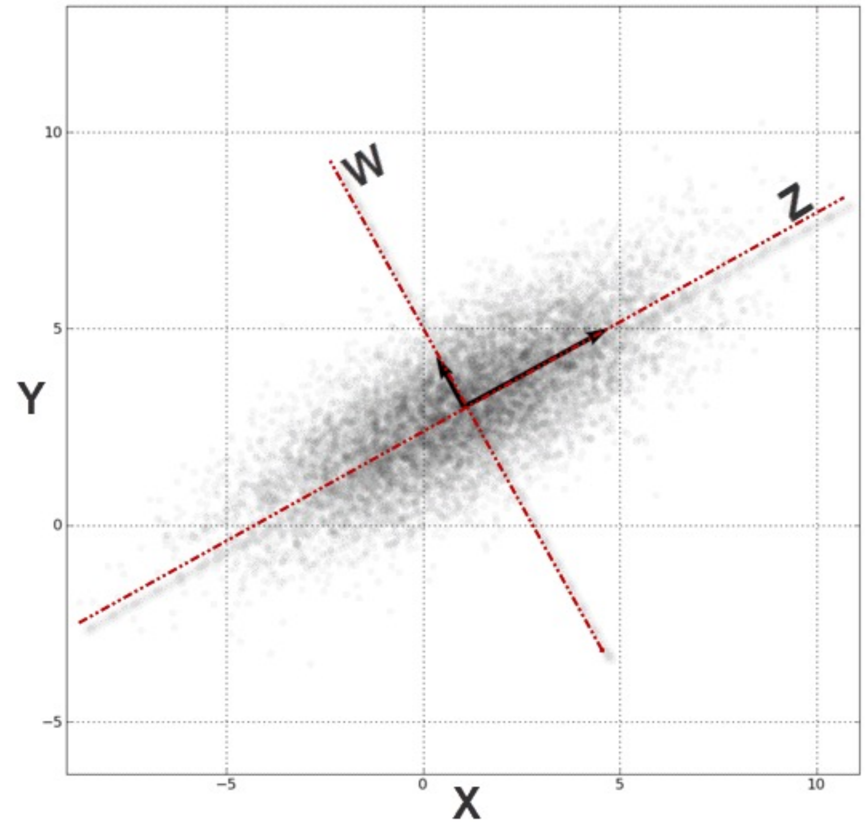
PCA and Neural Networks

- **Theorem** A neural net with the architecture <u>input-bottleneck-output</u> computes the same components as PCA
  - Even with non-linear activation functions
- ➢ Auto-encoders show similar behaviour to PCA



Preprocessing: Whitening

- Let $m = n$, i.e. #principal components = number of input features.
- Then PCA changes basis so that
    1. Each column has mean 0 and standard deviation 1.
    2. All covariances are 0.



Convolutional Auto-Encoder

Unsupervised Filter Learning

- Applying the (deterministic) auto-encoding idea with convolutions
  - ➢ Learn filters without supervision
- Basic idea is as with associative auto-encoders
  1. create a CNN with a bottleneck layer between encoding and decoding layers
  2. Loss function is squared error between input and output
- Problem: convolutions reduce image size
  - ➢ Cannot go from small encoded image to original image size

Solving the shrinkage problem

- Basic idea: if an image of size $n \times n$ is shrunk to $m \times m$, we pad the original image with enough 0s to make up for the loss

Original size	Shrinkage	Padded original	encoding	Reconstruction
n x n	m x m	n' x n'	m' x m'	n x n

Here n' > m' > n

Example Padding

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \end{array} \rightarrow \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 & 3 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 3 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 4 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 1 & 0 & 2 \end{array}$$

Figure 7.3: Padding an image for decoding in a convolutional AE

Conv2d_transpose handles the padding for you in Tensorflow

CN Auto-encoder example

	9	9		4	5	9	9
9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	7
8	8			9	9	9	
				9	9	5	
			6	9	9		
			8	9	9		
			9	9	7		
		1	9	9	1		
		4	9	9			
		1	9	4			

		6	6	2	2	5	5	6	6
		5	6	2	2	5	5	5	6
6	6	6	6	6	6	6	6	6	6
5	6	5	6	5	6	5	6	5	6
5	5	5	5	6	6	6	6		
5	5	5	5	5	6	5	6		
				6	6	6	6		
				5	6	5	6		
		1	1	6	6	4	4		
		1	1	5	6	4	4		
		2	2	6	6				
		2	2	5	6				

Figure 7.5: $14 * 14$ Minst 7, and version reconstructed from $7 * 7$ version

Conclusion

- General intuition for latent feature learning: A small set of unobserved features explains correlations among observed features.
- Finding a small set of explanatory features -> dimensionality reduction.
- Can visualize as merging feature columns.
- Autoencoders reduce dimensionality by reconstructing input from small number of hidden units
  - Think of each node in the bottleneck layer as representing an unobserved component
- Principal Component Analysis projects input to m-dimensional subspace