IT/PC/B/T/411

Machine Learning

Deep Learning Basics

Lecture 07: Factor Analysis

Dr. Pawan Kumar Singh
Department of Information Technology
Jadavpur University
pawankrsingh.cse@gmail.com
+91-6291555693

Supervised v.s. Unsupervised

Math formulation for supervised learning

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \boldsymbol{\mathcal{H}}$ that minimizes $\hat{L}(f) = \frac{1}{n} \sigma_{i=1}^{n} \; l(f, x_i, y_i)$
- s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

Unsupervised learning

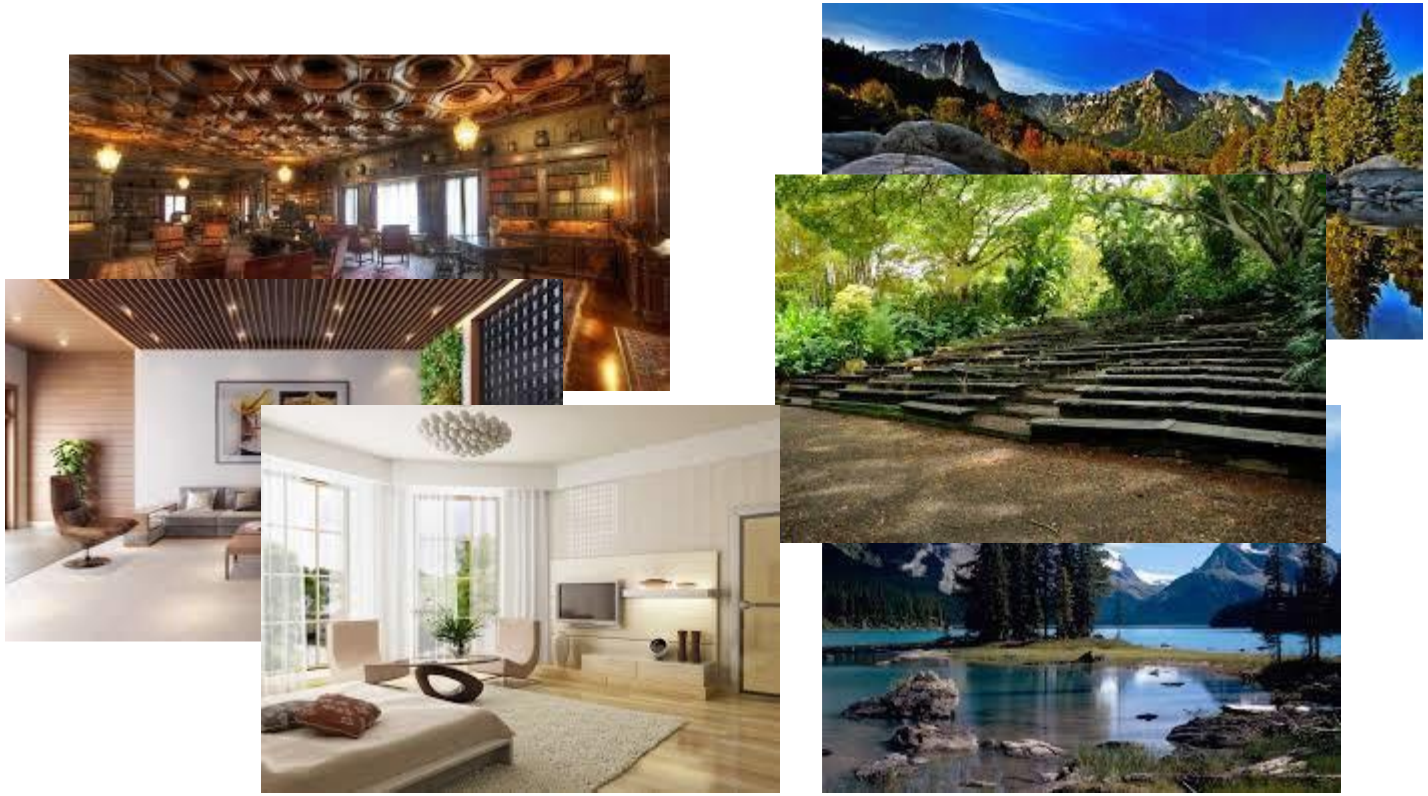
- Given training data $\{x_i : 1 \leq i \leq n\}$ i.i.d. from distribution D
- Extract some "structure" from the data

- Do not have a general framework
- Typical unsupervised tasks:
  - Summarization: clustering, dimension reduction
  - Learning probabilistic models: latent variable model, density estimation

Principal Component Analysis (PCA)

High dimensional data

- Example 1: images



Dimension: 300x300 = 90,000

High dimensional data

- Example 2: documents
- Features:
  - Unigram (count of each word): thousands
  - Bigram (co-occurrence contextual information): millions

- Netflix survey: 480189 users x 17770 movies

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	…
User 1	5	?	?	1	3	?	
User 2	?	?	3	1	2	5	
User 3	4	3	1	?	5	1	
…							

Example from Nina Balcan

Principal Component Analysis (PCA)

- Data analysis point of view: dimension reduction technique on a given set of high dimensional data $\{x_i : 1 \leq i \leq n\}$
- Math point of view: eigen-decomposition of the covariance (or singular value decomposition of the data)
- Classic, commonly used tool

Principal Component Analysis (PCA)

- Extract hidden lower dimensional structure of the data
  - Try to capture the variance structure as much as possible

- Computation: solved by singular value decomposition (SVD)

1hr.

Principal Component Analysis (PCA)

- **Definition:** an orthogonal projection or transformation of the data into a (typically lower dimensional) subspace so that the variance of the projected data is maximized.

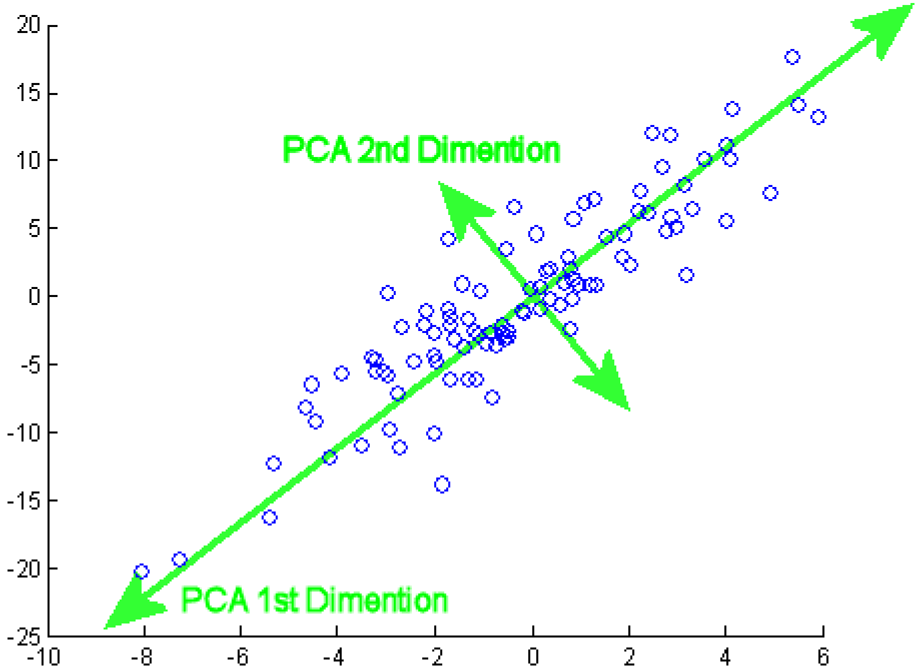


Figure from isomorphismes @stackexchange

Principal Component Analysis (PCA)

- An illustration of the projection to 1 dim
- Pay attention to the variance of the projected points

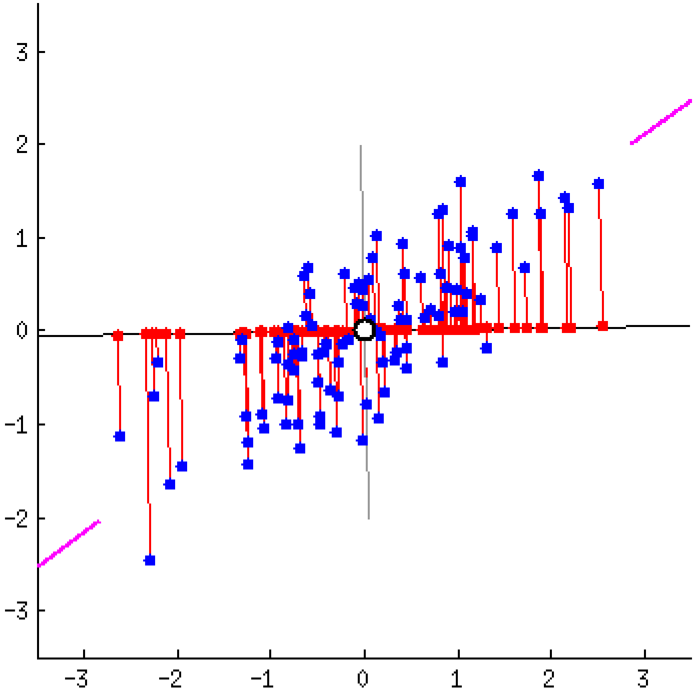


Figure from amoeba@stackexchange

V. V. I. P.

Principal Component Analysis (PCA)

- Principal Components (PC) are directions that capture most of the variance in the data
- First PC: direction of greatest variability in data
  - Data points are most spread out when projected on the first PC compared to any other direction

highest variance axis

- Second PC: next direction of greatest variability, orthogonal to first PC
- Third PC: next direction of greatest variability, orthogonal to first and second PC's
- …

Math formulation

- Suppose the data are centered: $\sigma_{i=1}^{n} x_i = 0$
- Then their projections on any direction v are centered: $\sigma_{i=1}^{n} v^{T} x_i = 0$

- First PC: maximize the variance of the projections

$$\max_{v} \overset{n}{\underset{i=1}{\fbox{?}}} (v^T x_i)^2, \qquad s.t.\;\; v^T v = 1$$

equivalent to

$$\max_{v} v^T X X^T v, \qquad s.t.\;\; v^T v = 1$$

where the columns of X are the data points

Math formulation

- First PC:

$$\max_{v} v^T X X^T v , \qquad s.t. \;\; v^T v = 1$$

where the columns of X are the data points

- Solved by Lagrangian: exists λ, so that

$$\max_{v} v^T X X^T v - \lambda v^T v$$

$$\frac{\partial}{\partial v} = 0 \;\rightarrow\; (XX^T - \lambda I)v = 0 \;\rightarrow\; XX^T v = \lambda v$$

Computation: Eigen-decomposition

- First PC: $XX^T v = \lambda v$

- XX^T : covariance matrix
- v : eigen-vector of the covariance matrix
- First PC: first eigen-vector of the covariance matrix

- Top k PC's: similar argument shows they are the top k eigen-vectors

Computation: Eigen-decomposition

- Top k PC's: the top k eigen-vectors $XX^T U = \Lambda U$

  where Λ is a diagonal matrix
- U are the left singular vectors of X

- Recall SVD decomposition theorem:
- An $m \times n$ real matrix M has factorization $M = U\Sigma V^T$ where U is an $m \times m$ orthogonal matrix, Σ is a $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ orthogonal matrix.

Equivalent view: low rank approximation

- First PC maximizes variance:

$$\max_{v} v^T X X^T v \,, \qquad s.t.\ \ v^T v = 1$$

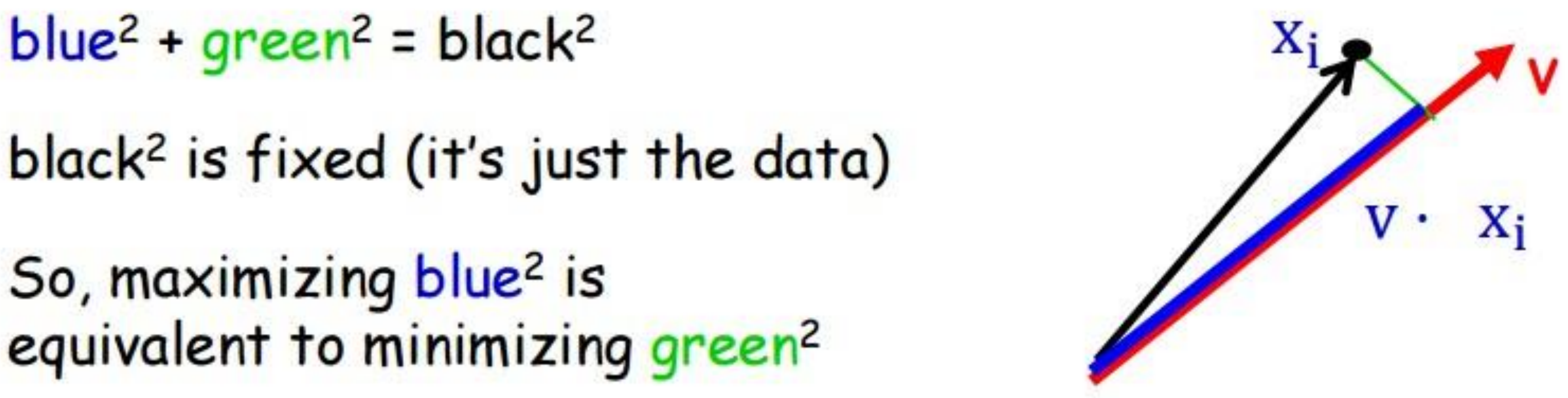
- Alternative viewpoint: find vector v such that the projection yields minimum MSE reconstruction

$$\min_{v} \ \frac{1}{n} \sum_{i=1}^{n} ||x_i - v v^T x_i||^2 \,, \qquad s.t.\ \ v^T v = 1$$

Equivalent view: low rank approximation

- Alternative viewpoint: find vector v such that the projection yields minimum MSE reconstruction

$$\min_{v} \frac{1}{n} \sum_{i=1}^{n} ||x_i - vv^T x_i||^2, \qquad s.t.\ \ v^T v = 1$$



blue² + green² = black²

black² is fixed (it's just the data)

So, maximizing blue² is equivalent to minimizing green²

Figure from Nina Balcan

Summary

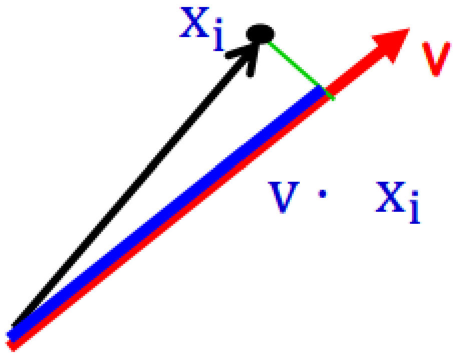
- PCA: orthogonal projection that maximizes variance
- Low rank approximation: orthogonal projection that minimizes error
- Eigen-decomposition/SVD

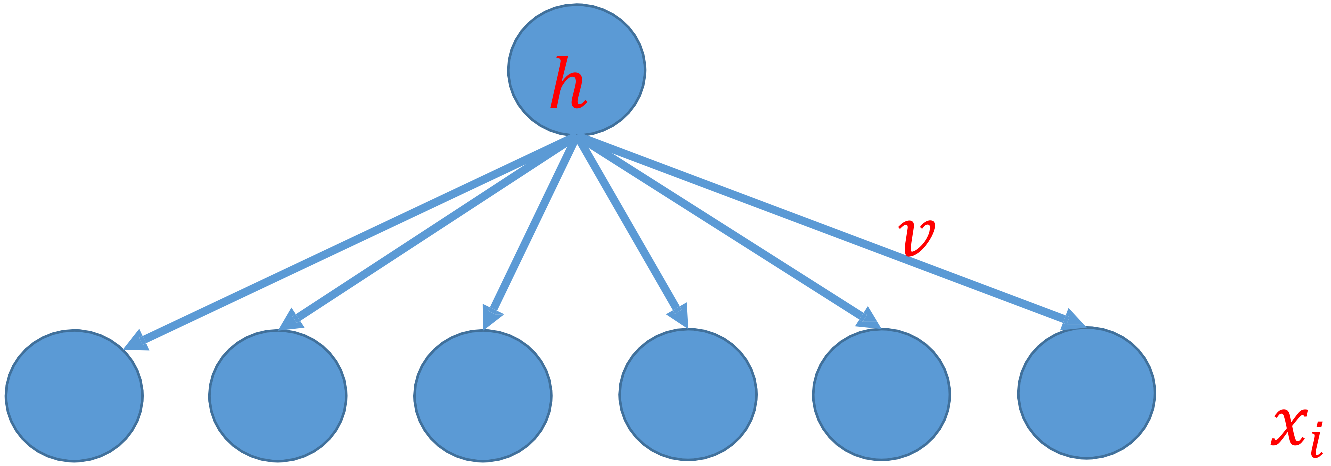
- All equivalent for centered data

Sparse coding

A latent variable view of PCA

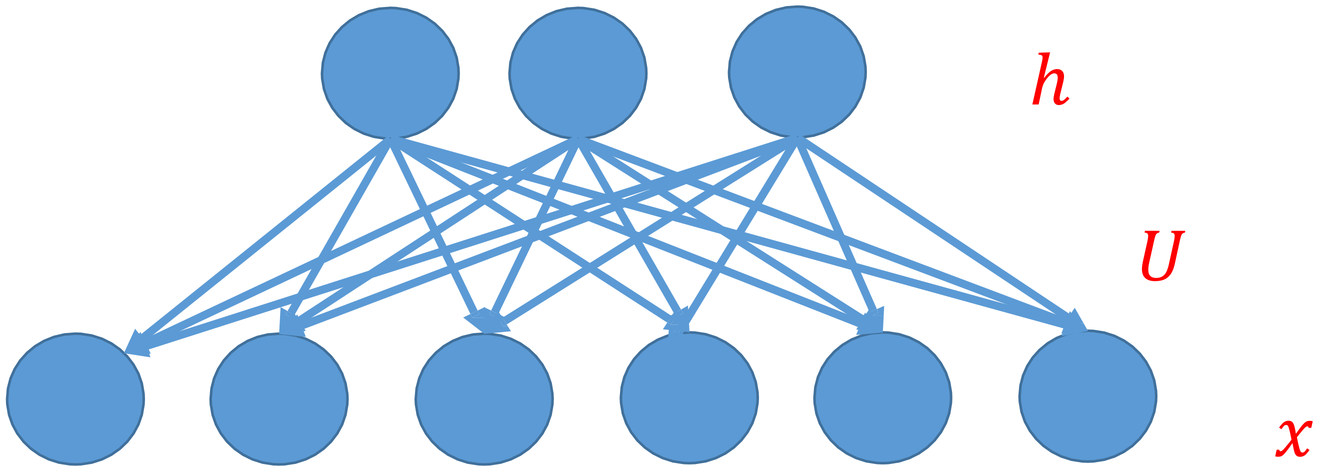
- Let $h_i = v^T x_i$
- Data point viewed as $x_i = v h_i + noise$





A latent variable view of PCA

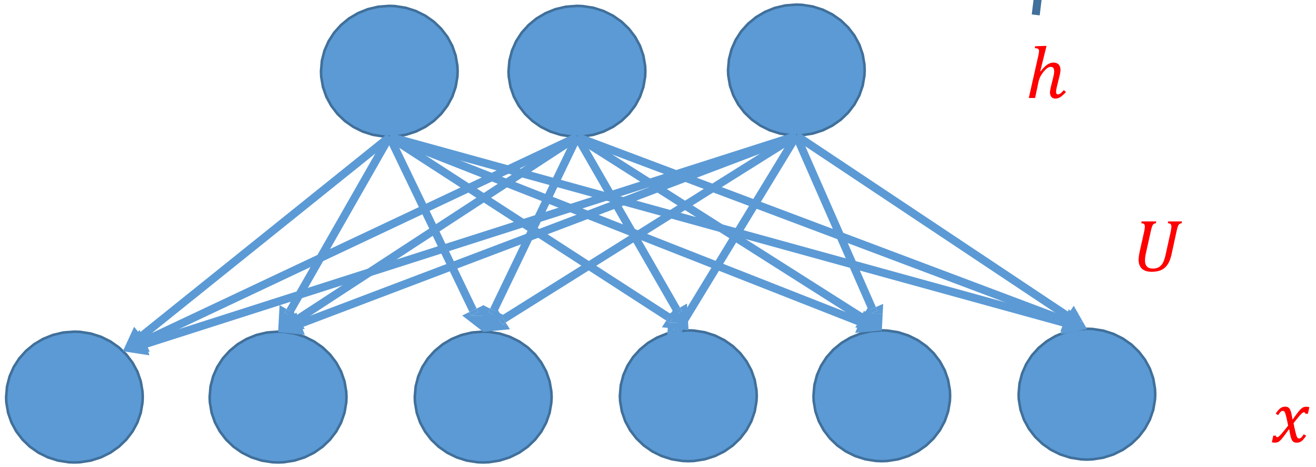
- Consider top k PC's U
- Let $h_i = U^T x_i$
- Data point viewed as $x_i = Uh_i + noise$



A latent variable view of PCA

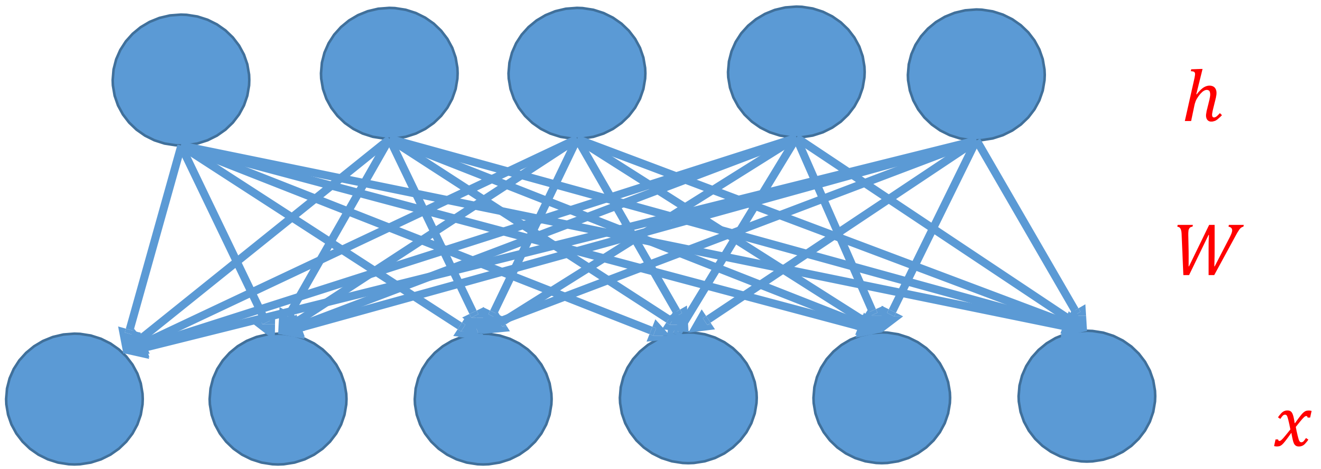
PCA structure assumption: h low dimension. What about other assumptions?

- Consider top k PC's U
- Let $h_i = U^T x_i$
- Data point viewed as $x_i = Uh_i + noise$



Sparse coding

- Structure assumption: h is sparse, i.e., $|h|_0$ is small
- Dimension of h can be large

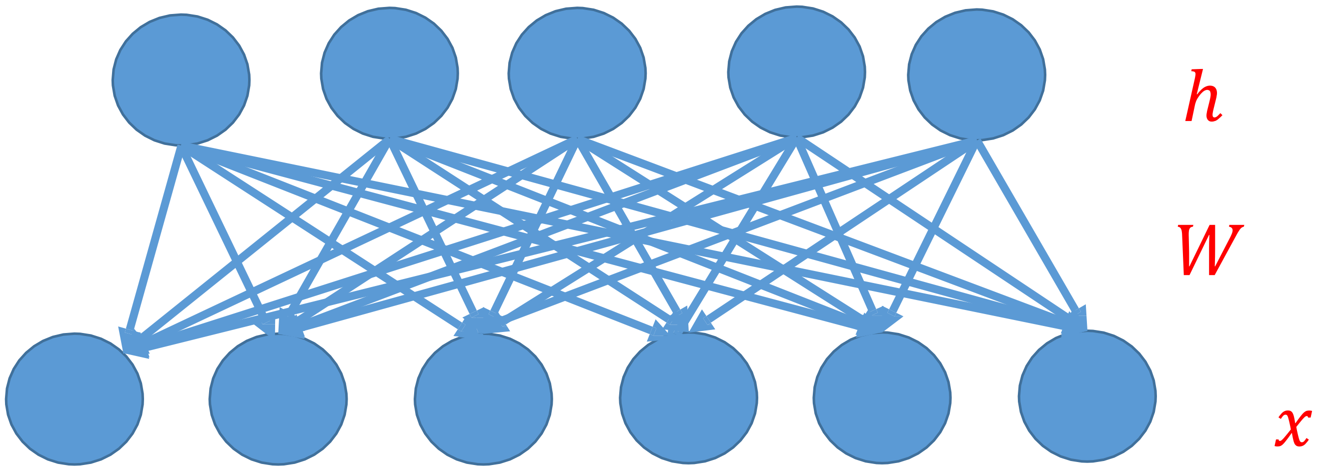


Sparse coding

- Latent variable probabilistic model view:

$$p(x|h) = Wh + N\left(0, \frac{1}{\beta} I\right), h \text{ is sparse,}$$

- E.g., from Laplacian prior: $p(h) = \frac{\lambda}{2}\exp(-\frac{\lambda}{2}|h|_1)$



Sparse coding

- Suppose W is known. MLE on h is

$$h^* = \arg\max_h \log p(h|x)$$

$$h^* = \arg\min_h \lambda ||h||_1 + \beta ||x - Wh||_2^2$$

- Suppose both W, h unknown.
    - Typically alternate between updating W, h

Sparse coding

- Historical note: study on visual system

- Bruno A Olshausen, and David Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." *Nature* 381.6583 (1996): 607-609.

Project paper list

Supervised learning

- AlexNet: ImageNet Classification with Deep Convolutional Neural Networks
- GoogLeNet: Going Deeper with Convolutions
- Residue Network: Deep Residual Learning for Image Recognition

Unsupervised learning

- Deep belief networks: A fast learning algorithm for deep belief nets
- Reducing the Dimensionality of Data with Neural Networks
- Variational autoencoder: Auto-Encoding Variational Bayes
- Generative Adversarial Nets

Recurrent neural networks

- Long-short term memory

- Memory networks

- Sequence to Sequence Learning with Neural Networks

You choose the paper that interests you!

- Need to consult with TA
    - Heavier responsibility on the student side if customize the project

- Check recent papers in the conferences ICML, NIPS, ICLR
- Check papers by leading researchers: Hinton, Lecun, Bengio, etc
- Explore whether deep learning can be applied to your application

- Not recommend arXiv: too many deep learning papers