Where are we...?

- We are now all proficient in understanding deep neural networks and how to optimize them
- But... many research frontiers in deep learning involve **probabilistic** models of the input $p_{model}(\mathbf{x})$
- We are often interested in using probabilistic inference to predict any of the variables in its environment, given any of the other variables

*** 99% of the material today is heavily borrowed from the Deep Learning textbook

Latent variables

- Many probabilistic models have latent variables, **h**, with

$$p_{model}(\mathbf{x}) = E_h p_{model}(\mathbf{x}|\mathbf{h})$$

- Latent variables are another way to represent the data
- Idea: distributed representations based on latent variables can obtain all of the advantages of learning which we have seen with deep networks

Latent variables: a review

- **Latent variables**, as opposed to observable variables, are variables that are not observed but instead inferred from observed variables
- Latent variable models are used in: psychology, economics, engineering, medicine, physics, ML/AI, bioinformatics, NLP, management, and pretty much everywhere else

Example of a latent variable

- In economics, we are often interested in measuring things such as quality of life, morale, happiness, and other things
- These things cannot be directly measured!
- The idea is to link these latent variables to observable variables
- For example, perhaps quality of life can be inferred from some linear combination of wealth, employment, environment, physical health, education, leisure time, etc...

Linear Factor Models

Back to deep learning...

- As an introduction to probabilistic models with latent variables, we start with one of the simplest classes: **linear factor models**
- Warning: you may not be implementing any linear factor models to solve state-of-the-art problems, **but** they provide a nice building block for mixture models or deeper probabilistic models
- Many of the approaches we discuss today are necessary to build generative models that more advanced deep models (keep coming to class!) models will expand upon

Linear factor models

- Defined by the use of a stochastic, linear decoder that generates $\mathbf{x}$ by adding noise to a linear transformation of $\mathbf{h}$
- Allow us to discover explanatory factors that have a simple joint distribution
- Simplicity of the linear decoder motivated these as some of the first latent variable models

Linear factor models (LFMs)

LFMs describe the data generation process as follows:

1. Sample the explanatory factors $\mathbf{h}$ from a distribution

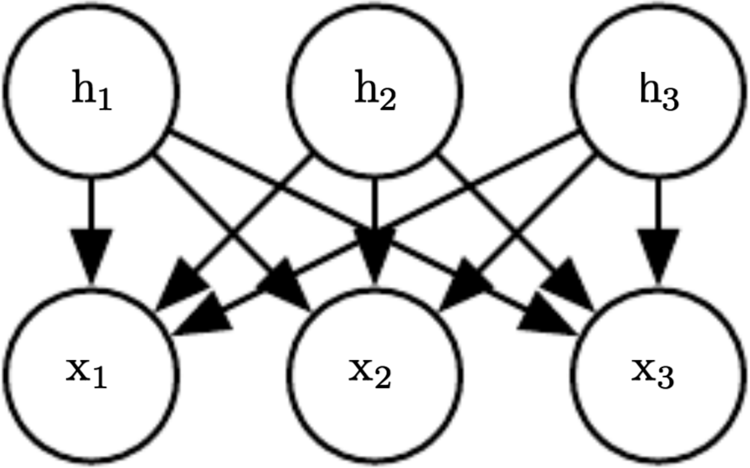
$$\mathbf{h} \sim p(\mathbf{h})$$

   where $p(\mathbf{h})$ is a factorial distribution (i.e. $p(\mathbf{h}) = \prod_i p(h_i)$)
2. Sample the real-valued observable variables given the factors:

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

   where the noise is typically Gaussian and diagonal

LFMs illustrated



$$\mathbf{x} = \boldsymbol{W}\,\mathbf{h} + \boldsymbol{b} + \textbf{noise}$$

Types of LFMs

- The directed graphical model on the previous slide describes the LFM family, where we assume that observed $\mathbf{x}$ is obtained by a linear combination of independent latent factors $\mathbf{h}$, plus some noise
- Different types of LFMs make different choices about the form of the noise and of the prior $p(\mathbf{h})$
- We will touch upon:
  - Probabilistic PCA and factor analysis
  - Independent component analysis (ICA)
  - Slow feature analysis
  - Sparse coding

Factor analysis

- (Batholomew, 1987; Basilevsky, 1994)
- Here, the latent variable prior is just the unit variance Gaussian:

$$\mathbf{h} \sim N(\mathbf{h}; \mathbf{0}, \mathbf{I})$$

- Observed values x_i are assumed to be **conditionally independent** given **h**
- That is, the noise is assumed to be drawn from a diagonal covariance Gaussian distribution, with covariance matrix $\psi = \text{diag}(\sigma_1^2, ..., \sigma_n^2)$
- The latent variables should **capture the dependencies** between the observed variables x_i
- Can show that **x** is a multivariate normal:

$$\mathbf{x} \sim N(\mathbf{x}; \mathbf{b}, \mathbf{W}\mathbf{W}^\mathsf{T} + \psi)$$

Probabilistic PCA

From factor analysis to probabilistic PCA

- A slight modification to the factor analysis model allows us to cast PCA in a probabilistic framework: make the conditional variances σ_i^2 equal to each other
- Now we have:

$$\mathbf{x} \sim N(\mathbf{x}; \mathbf{b}, \mathbf{W}\mathbf{W}^{\mathbf{T}} + \sigma^2 \mathbf{I}) \rightarrow \text{Factor Analysis}$$

- Equivalently:

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \sigma \mathbf{z} \rightarrow \text{probabilistic PCA}$$

where $\mathbf{z} \sim N(\mathbf{z}; \mathbf{0}, \mathbf{I})$ is Gaussian noise

- Can use an iterative EM algorithm to estimate $\mathbf{W}$ and σ^2 (Tipping and Bishop (1999))

Probabilistic PCA

- Probabilistic PCA takes advantage of the observation that most variations in the data can be captured by the latent variables, $\mathbf{h}$, up to some small residual **reconstruction error** σ^2
- Tipping and Bishop (1999) showed that probabilistic PCA becomes PCA as $\sigma \to 0$

Motivation behind probabilistic PCA

- In standard PCA, we assume linearity (bases of linear combinations of the measurement-basis), that large variances = import structure, and that principal components are orthogonal
- Linearity is not always justifiable!
- Calculating the covariance matrix can be very expensive in high-dimensional or big data settings
- De-correlation is not always the best approach (first and second order statistics are not always sufficient for revealing all dependencies in data, i.e. Gaussian data)

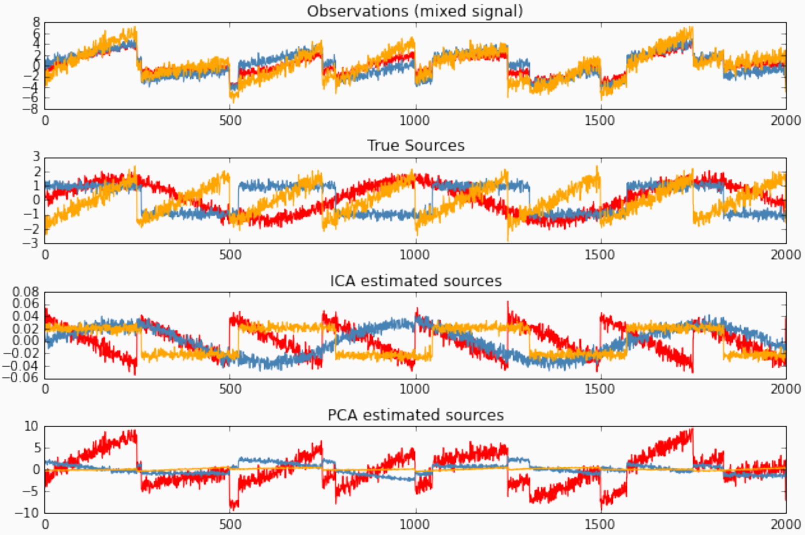
Independent Component Analysis

Independent component analysis (ICA)

- One of the oldest representation learning algorithms
- Models linear factors by seeking to separate an observed signal into underlying signals that are scaled and added together
- The underlying signals are intended to be fully independent
- $\exists$ many variants

ML@B

- A variant from Pham et al trains parametric generative model
- The prior $p(\mathbf{h})$ is fixed
- The model deterministically generates $\mathbf{x} = \mathbf{W}\mathbf{h}$
- A nonlinear change of variables allows us to determine $p(\mathbf{x})$
- Learning the model proceeds by using maximum likelihood


Observations (mixed signal)
True Sources
ICA estimated sources
PCA estimated sources
0
500
1000
1500
2000

Motivation behind ICA

- By choosing $p(\mathbf{h})$ to be independent, can recover factors that are as close as possible to independent
- Used to recover low-level signals that have been mixed
- Here, each data point x_i is one sensor's observation of the mixed signals, and each h_i is one estimate of the original signals
- Example: we have n people speaking simultaneously in n different microphones in different locations, ICA can detect changes in the volume between each speaker as heard by each microphone and separate the signals so that each h_i contains only one person speaking clearly

Applications of ICA

- Optical imaging of neurons
- Neuronal spike sorting
- Facial recognition
- Removing artifacts (i.e. eye blinks) from EEG (electroencephalography) data
- Predicting stock market prices
- Mobile phone communications