CS 4824/ECE 4424: Autoencoder

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.
Autoencoder

- Special type of feed forward network for
  - Compression
  - Denoising
  - Sparse representation
  - Data generation
Autoencoder

- Encoder: $f(\ )$
- Decoder: $g(\ )$
- Autoencoder: $g(f(x)) = x$
Linear Autoencoder

- $f$ and $g$ are linear
  - Matrix representation: $W_f$ and $W_g$

- Schematic
Linear Autoencoder

- **Objective**: find weights $W_f$ and $W_g$ that minimizes the reconstruction error

\[
\arg \min_{W} \frac{1}{2} \sum_{n} \| W_g W_f x_n - x_n \|_2^2
\]

- **Algorithm**: Backpropagation
  - Gradient descent

- **Hidden nodes**: compressed representation
Nonlinear Autoencoder

- $f$ and $g$ are nonlinear functions
  \[
  \text{arg min}_W \frac{1}{2} \sum_n \| g(f(x_n; W_f); W_g) - x_n \|^2_2
  \]

- Hidden nodes: nonlinear manifold
Deep Autoencoder
Deep Autoencoder

- $f$ and $g$ often consist of multiple layers

- In theory, one hidden layer in $f$ and $g$ is sufficient to represent any possible compression

- Multiple hidden layers in $f$ and $g$ is often better
Sparse Representations

- When more hidden nodes than inputs, use regularization to constrain autoencoder

- Example: force hidden nodes to be sparse
  \[
  \arg \min_{W} \frac{1}{2} \sum_{n} \left\| g(f(x_n; W_f); W_g) - x_n \right\|_2^2 + \text{cnnz}(f(x_n; W_f))
  \]
  - Where \( \text{cnnz}(f(x_n; W_f)) \) is the number of non-zero entries produced by \( f \)

- Approximate objective: L1 regularization
  \[
  \arg \min_{W} \frac{1}{2} \sum_{n} \left\| g(f(x_n; W_f); W_g) - x_n \right\|_2^2 + c \|f(x_n; W_f)\|_1
  \]
Denoising Autoencoder

- Consider noisy version $\tilde{x}$ of the input $x$
- Data denoising:

$$\arg\min_{W} \frac{1}{2} \sum_{n} \|g(f(\tilde{x}_n; W_f); W_g) - x_n\|_2^2 + c \|f(\tilde{x}_n; W_f)\|_1$$
Probabilistic Autoencoder

- Let $f$ and $g$ represent conditional distributions
  - $f : Pr(h \mid x; W_f)$ and $g : Pr(\tilde{x} \mid h; W_g)$
  - by using sigmoid, softmax or linear units at the hidden and output layers
- Schematic
Generative Model

- Sample $\mathbf{h}$ from some distribution $\Pr(\mathbf{h})$
- Sample $\mathbf{x}$ from the decoder: $\Pr(\mathbf{h} | \mathbf{x}; W_g)$