Large Scale Data Analysis Using Deep Learning

Autoencoder

U Kang
Seoul National University

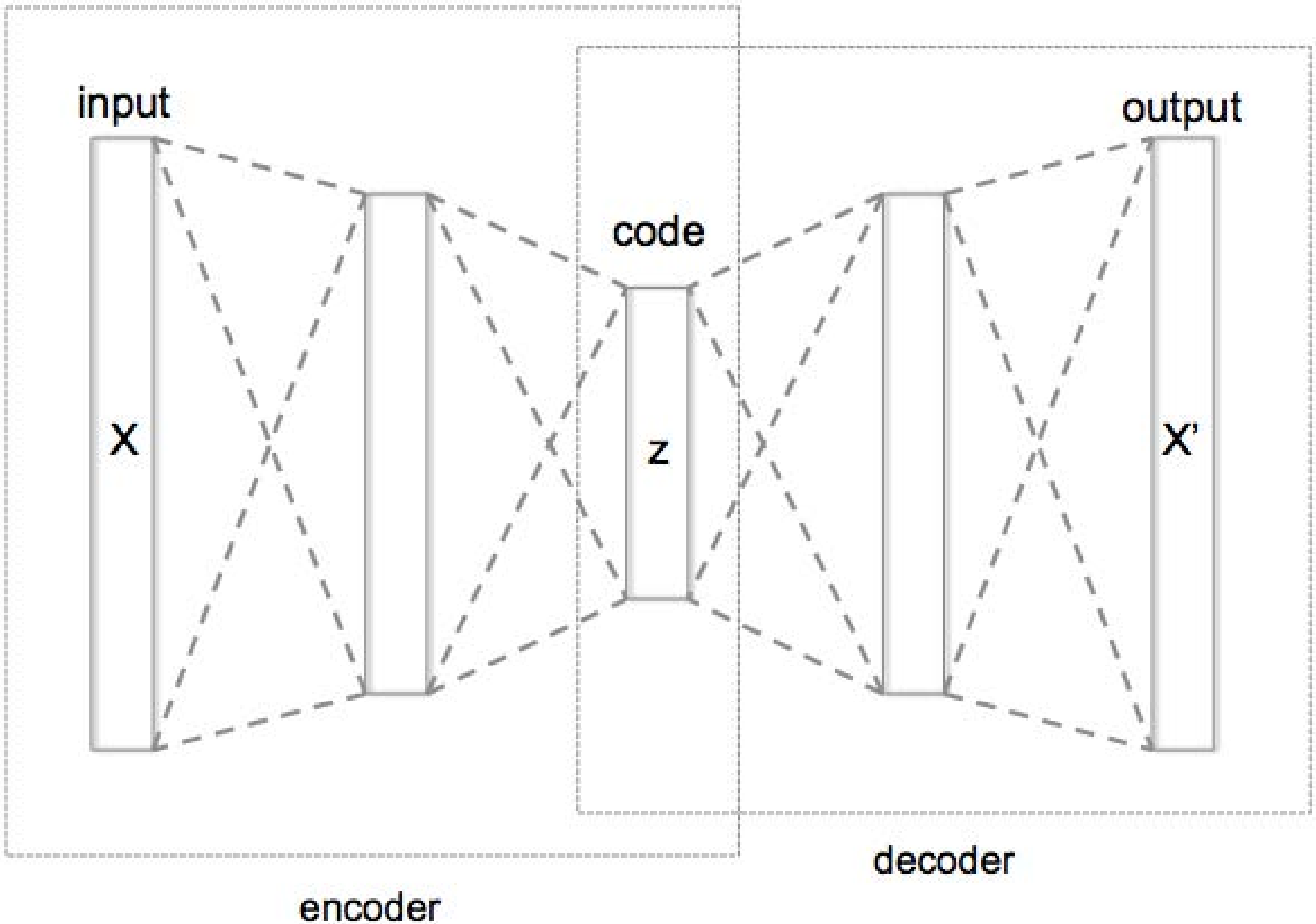
In This Lecture

- Autoencoder
  - Motivation
  - Undercomplete and overcomplet autoencoders
  - Regularization

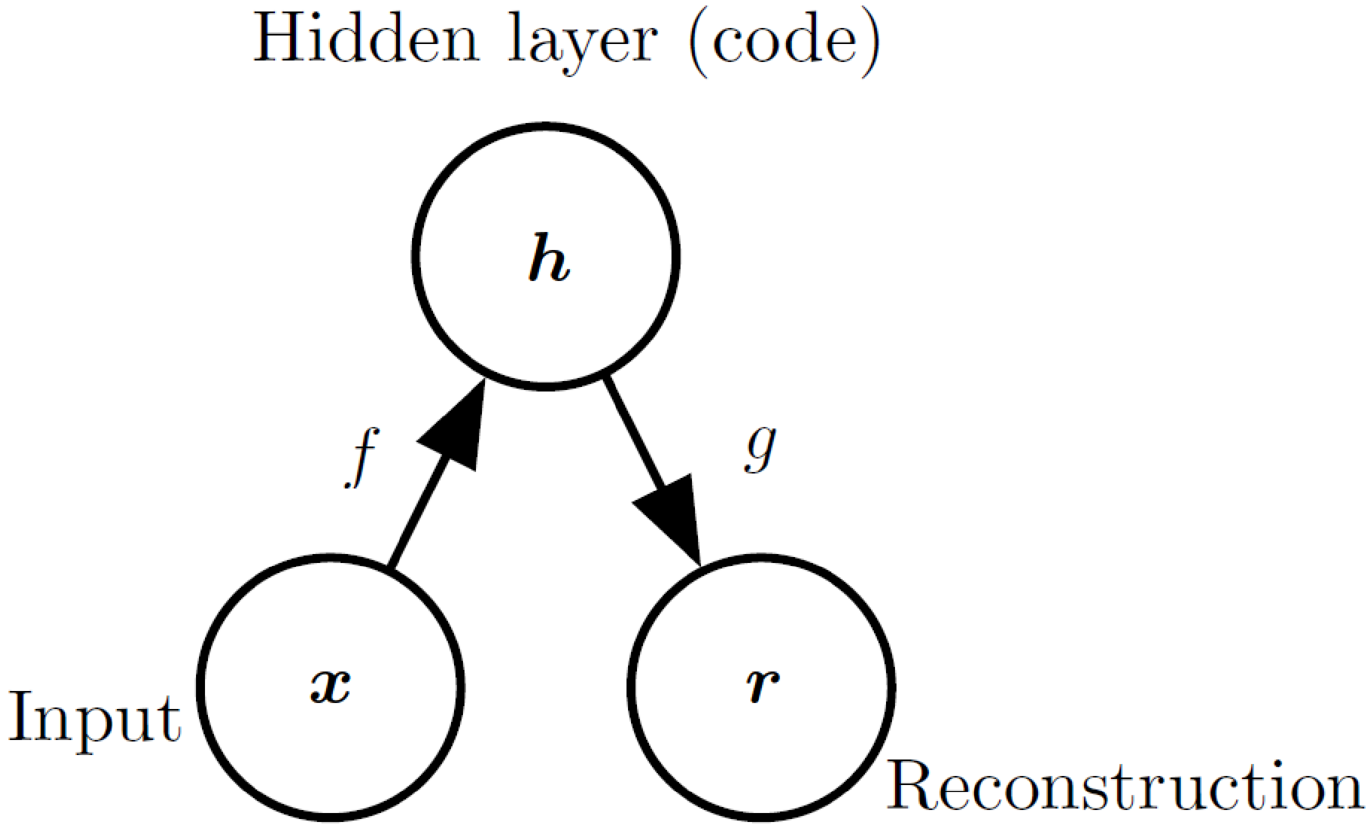
Autoencoder

- A neural network that is trained to attempt to copy its input to output
- Has a hidden layer h that describes a code used to represent the input
- Consists of two parts: an encoder function $h = f(x)$, and a decoder function $r = g(h)$ that reconstructs the original data
- The most simplest function would be an identity function for g and h; however, they are not useful to find important features of x
- Autoencoders are restricted to copy only approximately, and to copy only input that resembles the training data
  - This often leads to learn useful properties of data
- Can be thought of as a dimensionality reduction

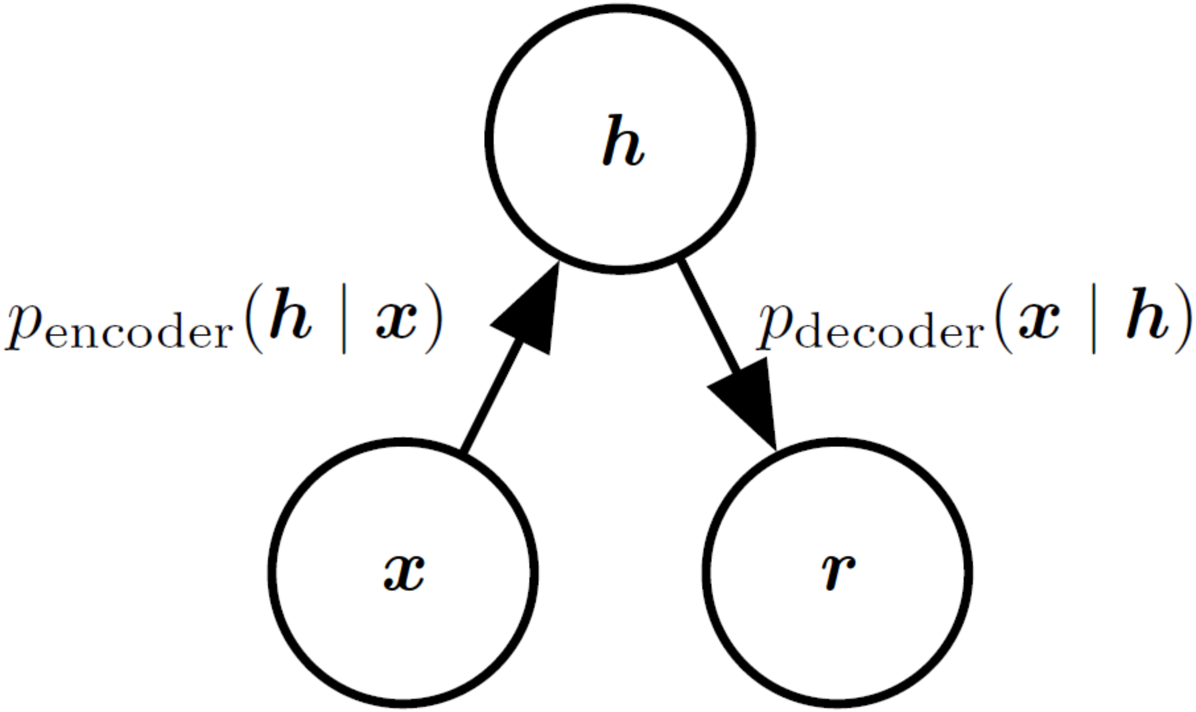
Autoencoder



Structure of an Autoencoder



Stochastic Autoencoders



Undercomplete Autoencoders

- Copying the input to the output seems useless
- We are not typically interested in the output of the decoder; we hope that training the autoencoder to perform the copying task will result in h taking on useful properties
- Undercomplete autoencoder
  - h has smaller dimension than x; this allows to learn the most salient features of the data distribution
  - Learning process: minimizing a loss function L(x, g(f(x)))
  - When the decoder is linear and L is the mean square error, an undercomplete autoencoder learns to span the same subspace as PCA
  - Autoencoders with nonlinear encoder and decoder functions learn a more powerful nonlinear generalization of PCA
  - Undercomplete autoencoders fail to learn anything useful if the encoder and decoder are given too much capacity: it can learn to perform the copying task without extracting useful information about the distribution of the data

Regularized Autoencoders

- Undercomplete autoencoders fail to learn anything useful if the encoder and decoder are given too much capacity
- A similar problem occurs if the hidden code is allowed to have dimension equal to the input
  - Overcomplete case: hidden code has dimension greater than the input
- In these cases, authoencoder can learn to copy input to output, without learning anything useful
- Regularized autoencoder: rather than limiting the model capacity (shallow encoder/decoder, and small code size), use a loss function that encourages the model to learn useful features
  - Sparse autoencoders
  - Denoising autoencoders
  - Contractive autoencoders
  - Autoencoders with dropout on the hidden layer

Sparse Autoencoders

- Limit capacity of autoencoder by adding a term to the cost function penalizing the code for being larger
  - $L\left(x, g\left(f(x)\right)\right) + \Omega(h)$
    where $\Omega(h) = \lambda \sum_i |h_i|$
  - By limiting the code h, autoencoders learn unique and important features

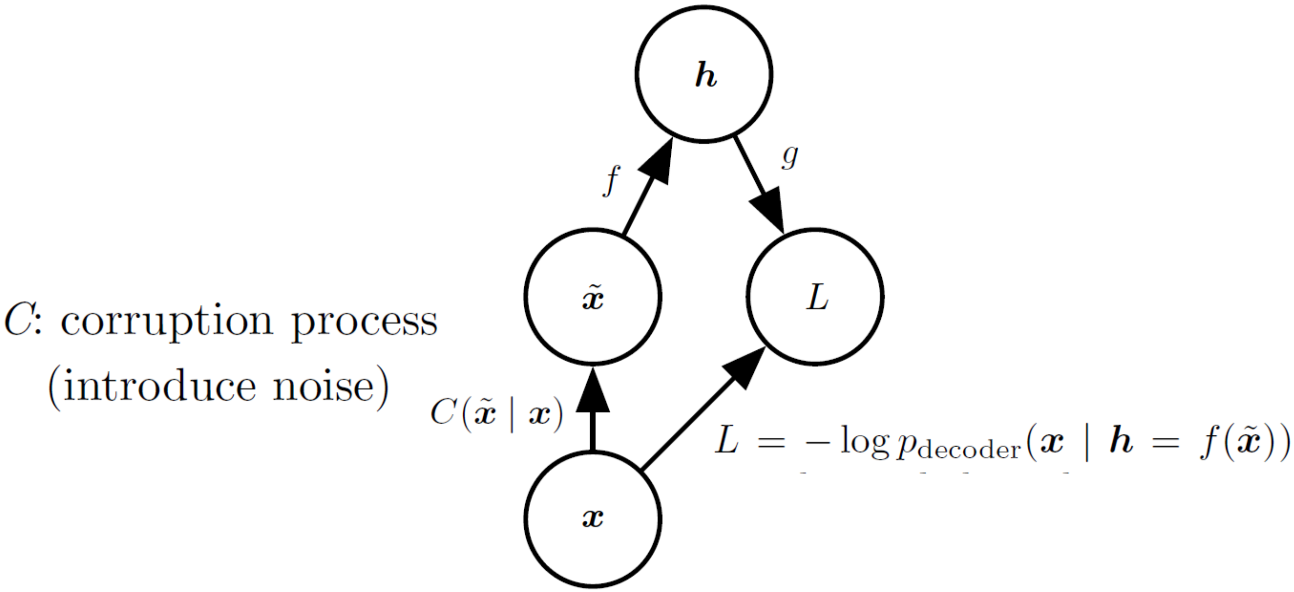
Denoising Autoencoder

- Rather than adding a penalty Ω to the cost function, we can obtain an autoencoder that learns something useful by changing the reconstruction error term
- Typical autoencoders minimize $L\left(x, g\left(f(x)\right)\right)$
- Denoising autoencoder (DAE) minimizes $L\left(x, g\left(f(\tilde{x})\right)\right)$

  where $\tilde{x}$ is a copy of x with some noise or corruption
- Denoising autoencoders must therefore undo this corruption rather than simply copying the input

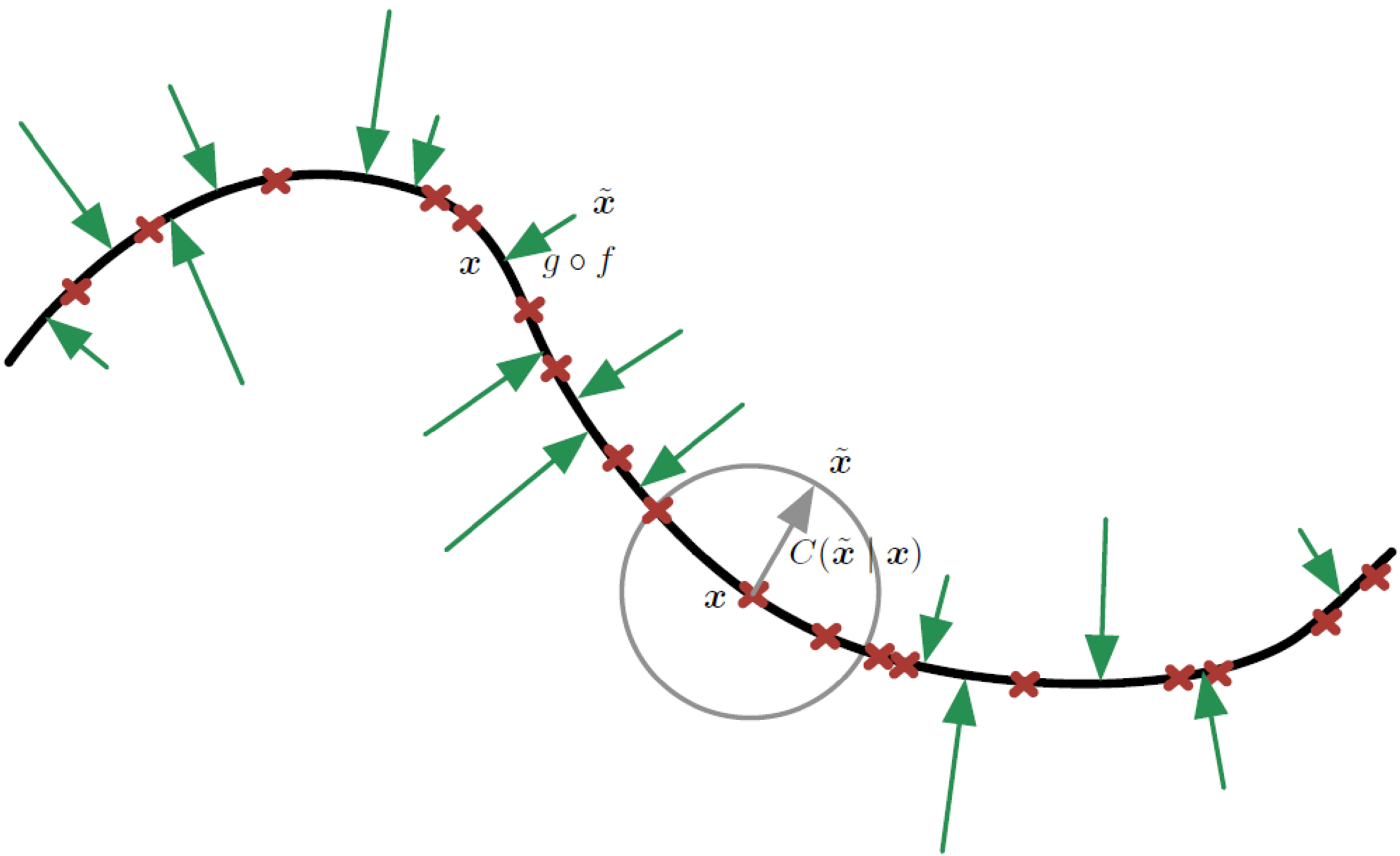
Denoising Autoencoder

- DAE training procedure
  - Sample a training example x from the training data
  - Sample a corrupted version $\tilde{x}$ from C($\tilde{x}|x$) where C is a conditional distribution of corrupted samples $\tilde{x}$ given a data sample x
  - Use $(x, \tilde{x})$ as a training example for estimating the autoencoder reconstruction distribution $p_{decoder}(x|h)$ where h is the output of the encoder $f(\tilde{x})$

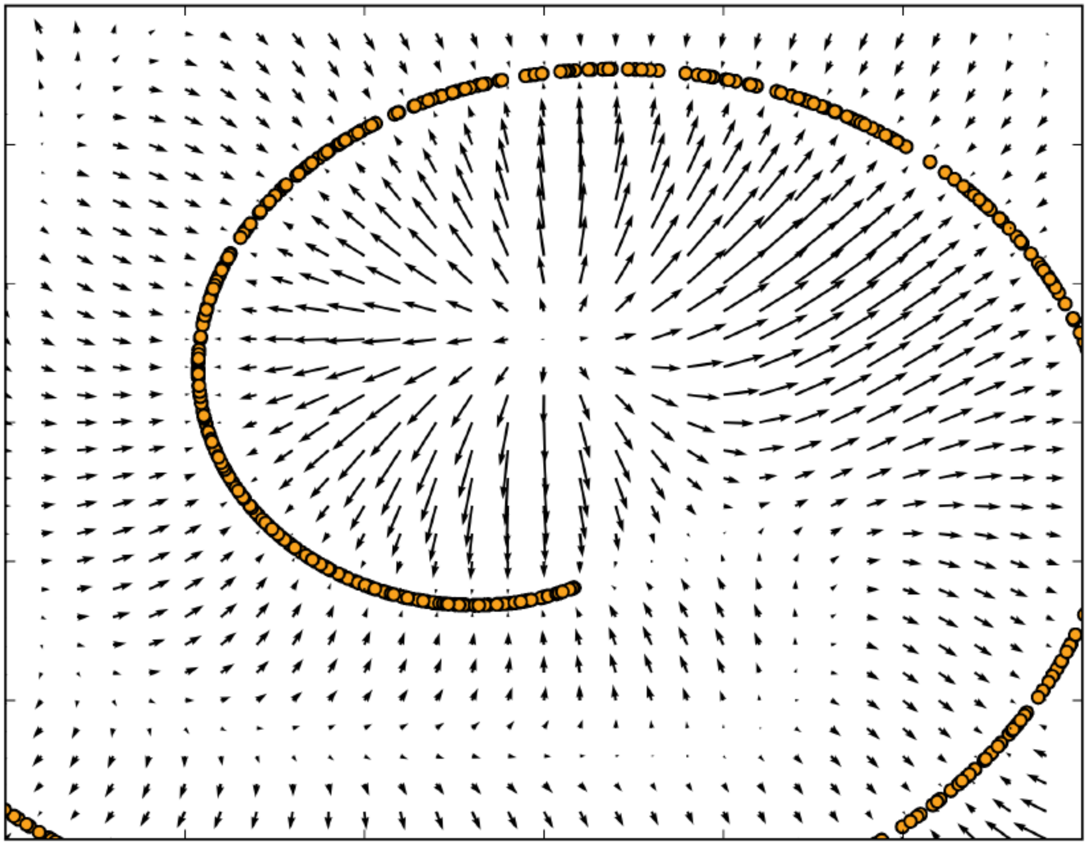


Denoising Autoencoders Learn a Manifold

- DAE maps each data point to its nearest point on the manifold



Vector Field Learned by a Denoising Autoencoder



Contractive Autoencoder

- As in sparse autoencoder, use a penalty term Ω, but with a different form
  - $L\left(x, g\left(f(x)\right)\right) + \Omega(h, x)$
    where $\Omega(h, x) = \lambda \sum_i ||\nabla_x h_i||^2$
- This forces the model to learn a function that does not change much when x changes slightly
  - For an "identity" encoder, the penalty would be large
- Connection between DAE and contractive autoencoder
  - For a small Gaussian input noise, the denoising reconstruction error is equivalent to a contractive penalty

Representational Power, Layer Size and Depth

- Autoencoders are often trained with only a single layer encoder and a single layer decoder
- However, deep encoders and decoders offer many advantages
  - Because autoencoders are feedforward networks
  - Depth can exponentially reduce the computational cost of representing some functions
  - Depth can also exponentially decrease the amount of training data needed to learn some functions
- A common strategy for training a deep autoencoder is to greedily pretrain the deep architecture by training a stack of shallow autoencoders
  - Thus, we often encounter shallow autoencoders even in the case of a deep autoencoder

What you need to know

- Autoencoder
  - Motivation
    - Learn low dimensional embedding of data points, by learning to reconstruct output given input
  - Undercomplete and overcomplete autoencoders
    - Undercomplete autoencoders avoid learning trivial function, but with low capacity
    - Overcomplte autoencoders can avoid learning trivial function via regularization
  - Regularization
    - Sparse, denoising, contractive autoencoders

Questions?