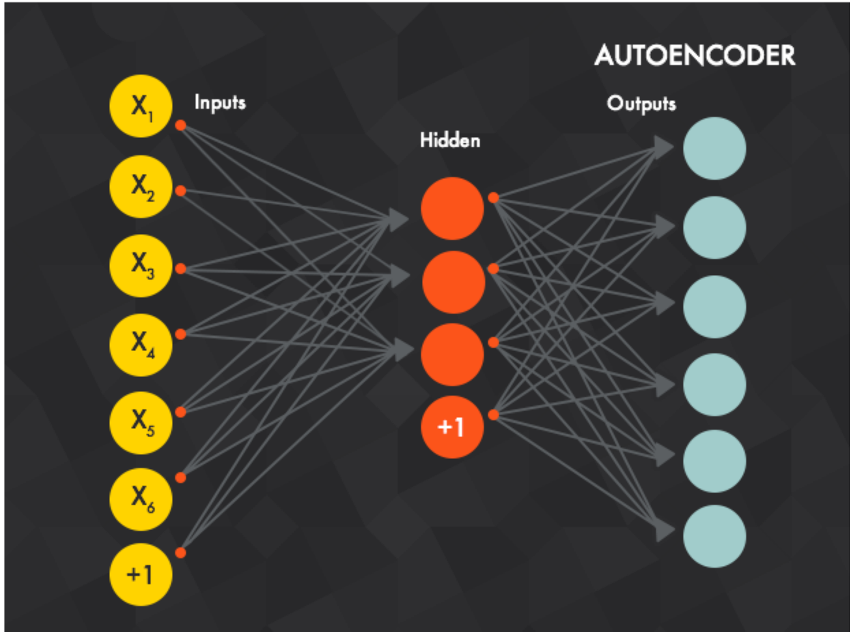
Unsupervised autoencoders for anomaly detection

noise, sparseness, stacks

Autoencoder

- Unsupervised Neural Network for learning efficient codings (Representations)
- Train by minimizing reconstruction error



Loss function

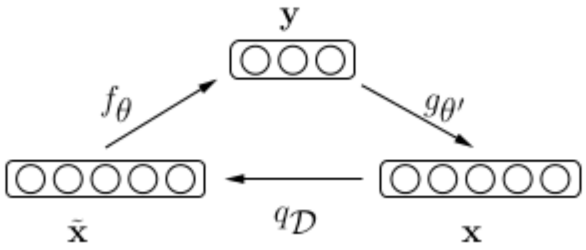
- Reconstruction Error

$$\mathbf{y} = f_\theta(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \qquad \theta = \{\mathbf{W}, \mathbf{b}\}$$

$$\mathbf{z} = g_{\theta'}(\mathbf{y}) = s(\mathbf{W}'\mathbf{y} + \mathbf{b}') \qquad \theta' = \{\mathbf{W}', \mathbf{b}'\}$$

$$\begin{aligned} \theta^\star, \theta'^\star &= \arg\min_{\theta,\theta'} \frac{1}{n} \sum_{i=1}^{n} L\left(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}\right) \\ &= \arg\min_{\theta,\theta'} \frac{1}{n} \sum_{i=1}^{n} L\left(\mathbf{x}^{(i)}, g_{\theta'}(f_\theta(\mathbf{x}^{(i)}))\right) \end{aligned}$$

- Best result?
- Identity Matrix : $W' = W^{-1}$



Overcoming Identity matrices

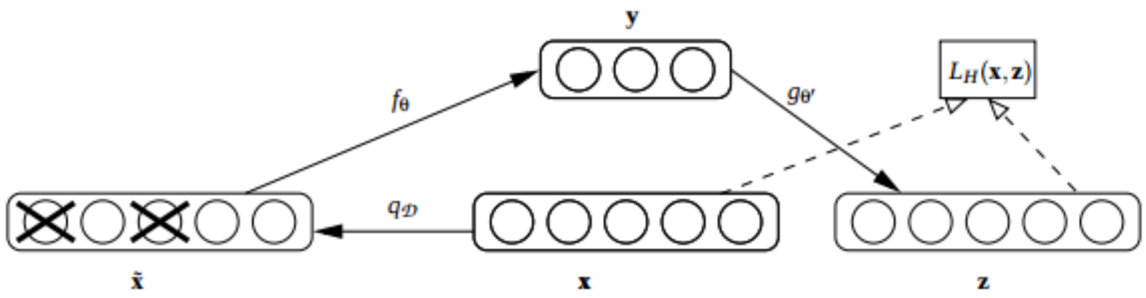
- To get interesting representations
  - Use bottlenecks (Undercomplete representations)
  - Use sparse coding (Overcomplete representations)

- Bottlenecks
  - dim(hidden) < dim(input)
- Sparse coding
  - dim(hidden) > dim(input)
  - active(hidden) : sparse

- Sparsity and bottlenecks forces only useful data to pass through

Robustness

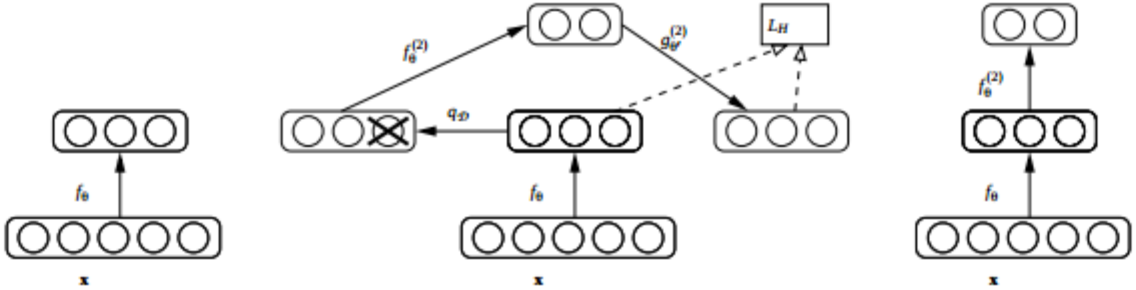
- Assumption
  - a good representation is one that can be obtained robustly from a corrupted input and that will be useful for recoering the corresponding clean input
- Denoising autoencoder
  - reconstruct the input from a corrupted input



Adding Stacks

- Assumption
  - higher levels of features can give light to higher level dependencies of the given inputs

- Stacking hierarchies of layers can give more interesting representations



Anomaly Detection

- Using autoencoders
    - By use of reconstruction errors, find out anomalies of the data
    - Investigate the represenations learned in the layers to find out the dependcies of inputs
- Use different schemes
    - RBM
    - Autoencoder
    - Sparse coding