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**

\[
\begin{align*}
  y &= f_\theta(x) = s(Wx + b) \quad \theta = \{W, b\} \\
  z &= g_{\theta'}(y) = s(W'y + b') \quad \theta' = \{W', b'\}
\end{align*}
\]

\[
\begin{align*}
  \theta^*, \theta'^* &= \arg\min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^{n} L\left(x^{(i)}, z^{(i)}\right) \\
  &= \arg\min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^{n} L\left(x^{(i)}, g_{\theta'}(f_\theta(x^{(i)}))\right)
\end{align*}
\]

- **Best result?**
- **Identity Matrix**: \( W' = W^{-1} \)
Overcoming Identity matrices

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

- Bottlenecks
  - $\dim(\text{hidden}) < \dim(\text{input})$

- Sparse coding
  - $\dim(\text{hidden}) > \dim(\text{input})$
  - $\text{active(\text{hidden})} : \text{sparse}$

- 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 recovering 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 representations learned in the layers to find out the dependencies of inputs

- **Use different schemes**
  - RBM
  - Autoencoder
  - Sparse coding