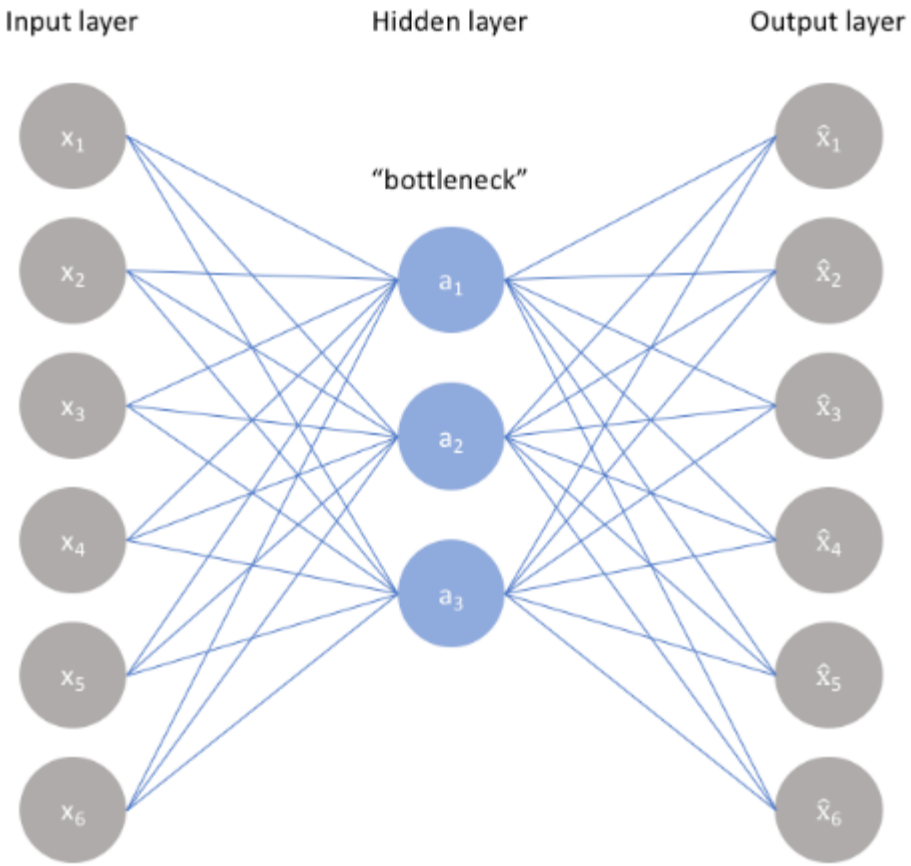
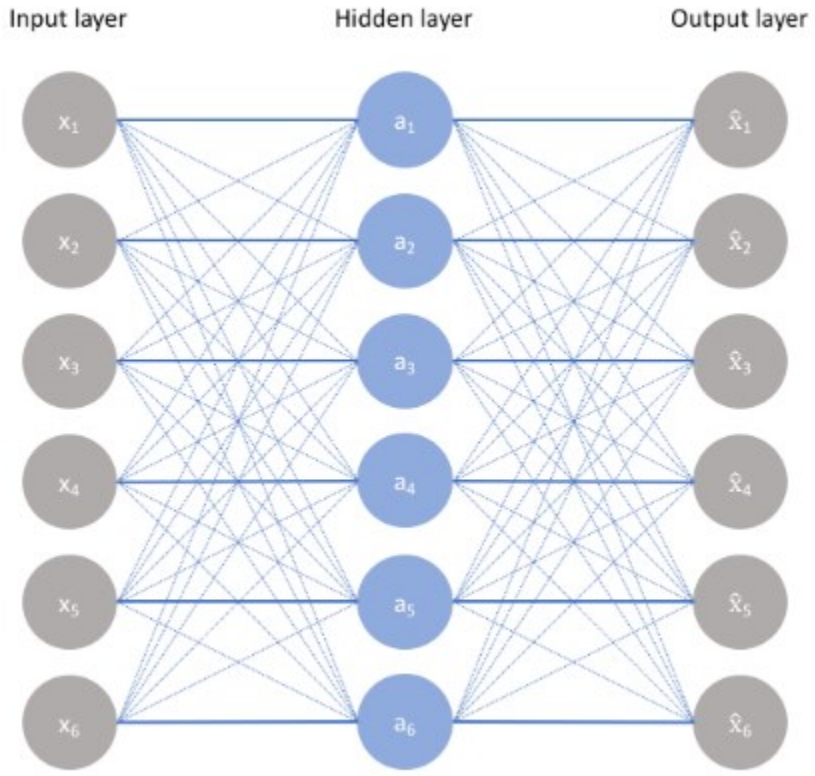
Introduction to Autoencoders

- Autoencoders are an unsupervised learning technique
- Autoencoders leverage neural networks for representation learning.
- Autoencoders introduce a bottleneck which forces a compressed knowledge representation of the original input.
- The idea is to capture a structure in the data if one exists in the bottleneck. Once identified, the underlying structure can be used for other things like anomaly detection.

- First, we take an unlabeled dataset and attempt to reconstruct the original (x) by obtaining its reconstruction(x^). This network can be trained by minimizing the reconstruction error.
- As the name implies, the reconstruction error represents the difference between the original and the reconstruction.
- Note that the reconstruction occurs through a bottleneck.
- A bottleneck constrains the amount of information that can traverse the full network, forcing a learned compression of the input data.

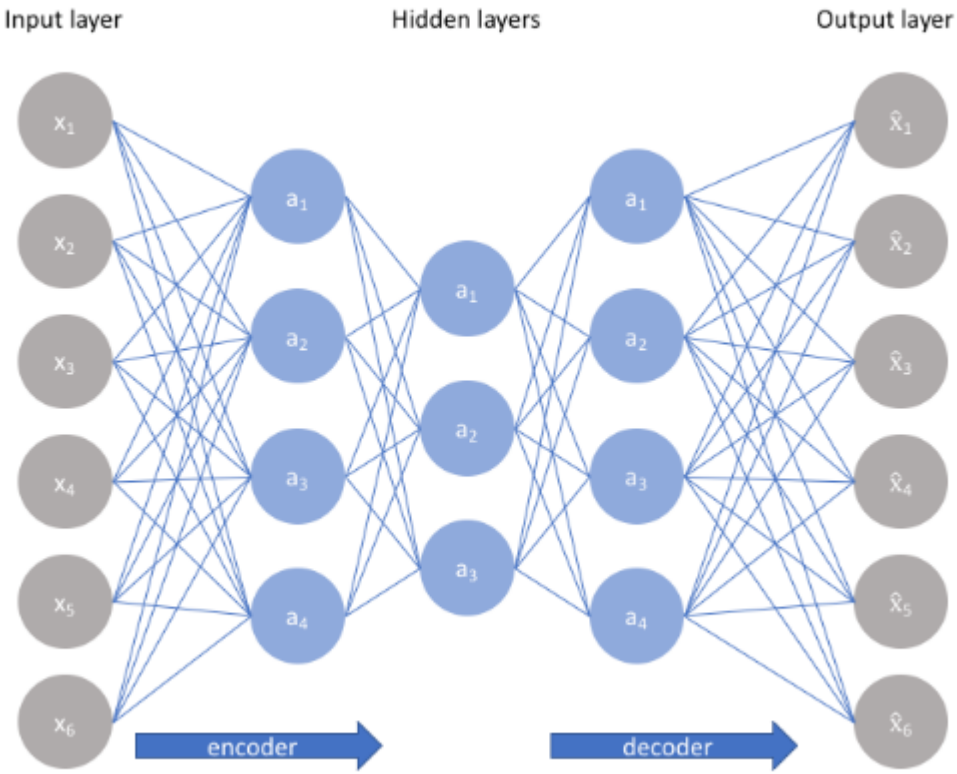

Input layer
Hidden layer
Output layer
"bottleneck"
x_1
x_2
x_3
x_4
x_5
x_6
a_1
a_2
a_3
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$


Input layer
Hidden layer
Output layer
x_1
x_2
x_3
x_4
x_5
x_6
a_1
a_2
a_3
a_4
a_5
a_6
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$

The ideal autoencoder manages a trade-off between two things

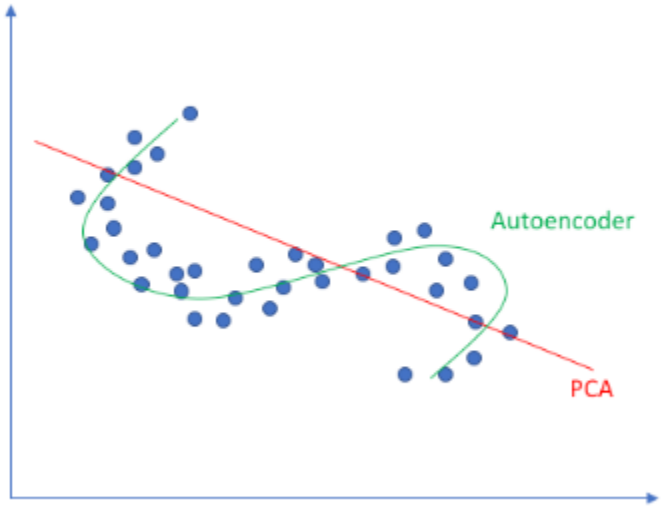
- Sensitivity to inputs to be able to create a reconstruction
- Insensitivity to inputs to avoid memorizing the training data.
- This involves constructing a loss function which is a representation error and a regularizer

$$\mathcal{L}\left(x, \hat{x}\right) + regularizer$$


Input layer
Hidden layers
Output layer
x1
x2
x3
x4
x5
x6
a1
a2
a3
a4
a1
a2
a3
a1
a2
a3
a4
x̂1
x̂2
x̂3
x̂4
x̂5
x̂6
encoder
decoder

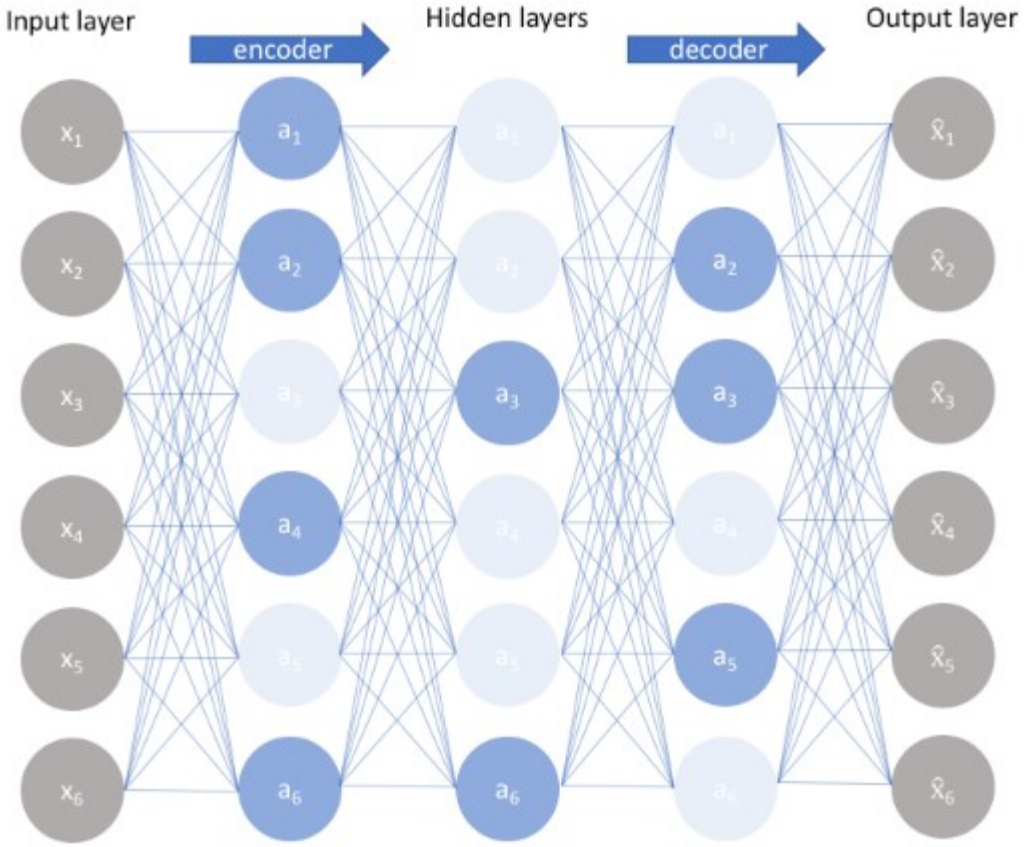
- Because neural networks are capable of learning nonlinear relationships, this can be thought of as a more powerful (nonlinear) generalization of PCA.
- Whereas PCA attempts to discover a lower dimensional hyperplane which describes the original data, autoencoders are capable of learning nonlinear manifolds (a manifold is defined in simple terms as a continuous, non-intersecting surface). The difference between these two approaches is visualized below.

Linear vs nonlinear dimensionality reduction



Sparse autoencoders

- Sparse autoencoders offer us an alternative method for introducing an information bottleneck without requiring a reduction in the number of nodes at our hidden layers.
- Instead of the bottleneck layer, we will introduce an artificial scarcity by learning encoding and decoding from a small number of neurons(spare autoencoding).
- In the diagram of a generic sparse autoencoder, the opacity of a node corresponds with the level of activation.


Input layer
encoder
Hidden layers
decoder
Output layer
x_1
x_2
x_3
x_4
x_5
x_6
a_1
a_2
a_3
a_4
a_5
a_6
a_1
a_2
a_3
a_4
a_5
a_6
a_1
a_2
a_3
a_4
a_5
a_6
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$