Autoencoder

What is autoencoder?

A: It is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back **from** the reduced encoded representation **to** a representation that is as close to the original input as possible.

It can be used to reduce the data into a low dimensional latent space

Credit: https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726

Autoencoder

Why we need autoencoder?

A: For **dimensionality reduction.**

Why we want to do dimensionality reduction?

A: In order to catch useful indicators and obtain a more accurate result, we tend to add as many features as possible at first. However, after a certain point, the performance of the model will decrease with the increasing number of elements, which is referred as **"The Curse of Dimensionality."**

The curse of dimensionality occurs because the sample density decreases exponentially with the increase of the dimensionality. When we keep adding features without increasing the number of training samples as well, the dimensionality of the feature space grows and becomes sparser and sparser. Due to this sparsity, it becomes much easier to find a "perfect" solution for the machine learning model which highly likely leads to overfitting.

Credit: https://medium.com/@cxu24/why-dimensionality-reduction-is-important-dd60b5611543

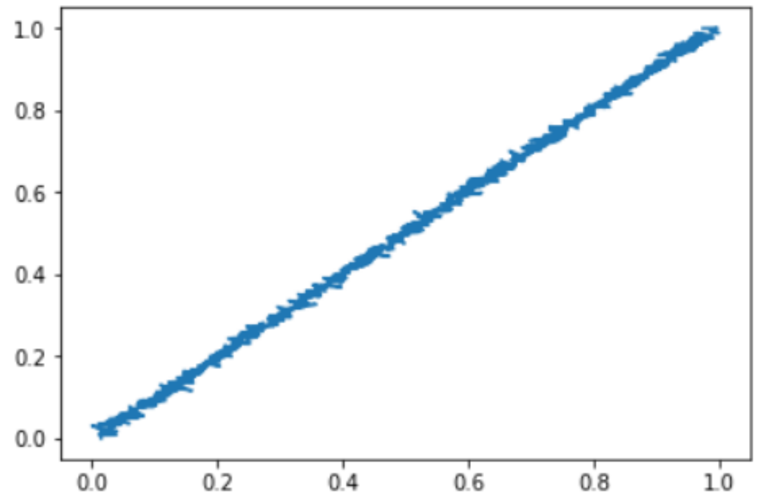
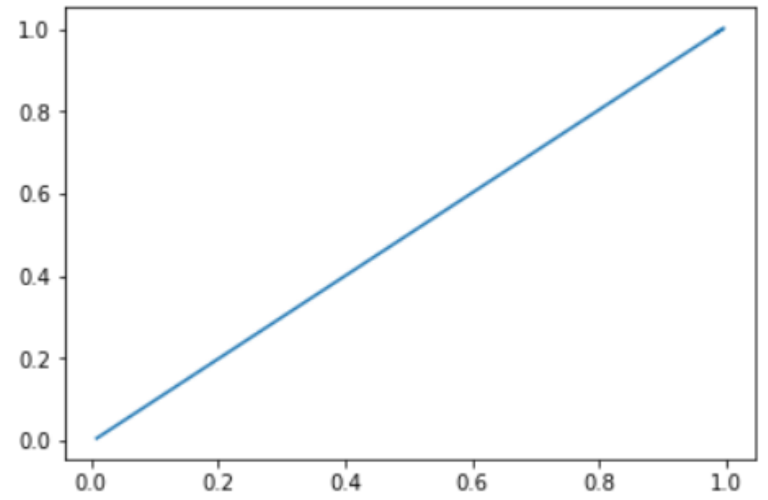
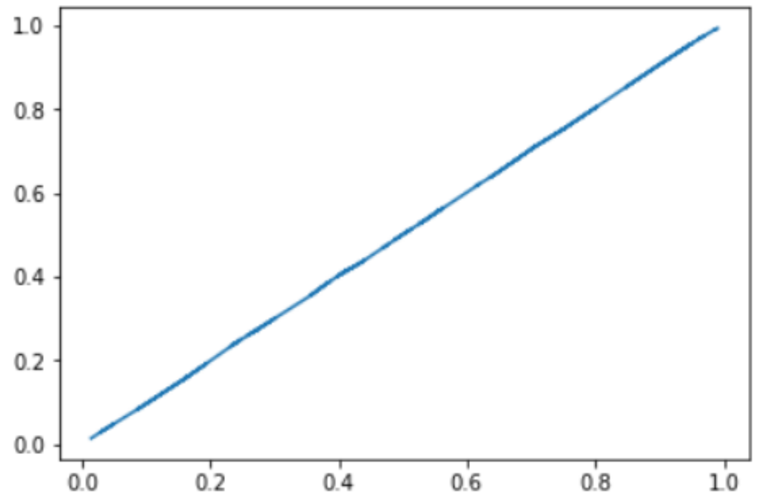
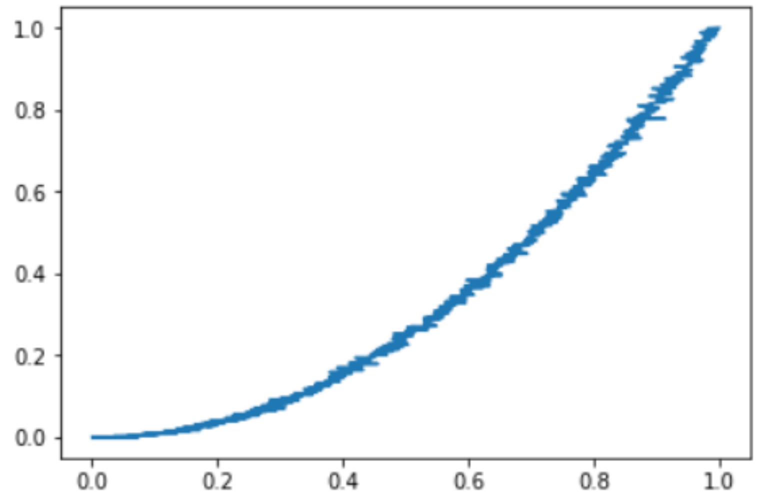
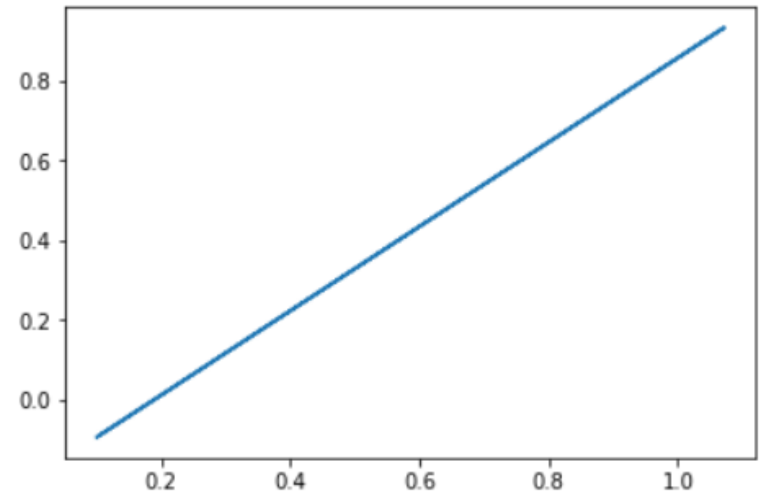
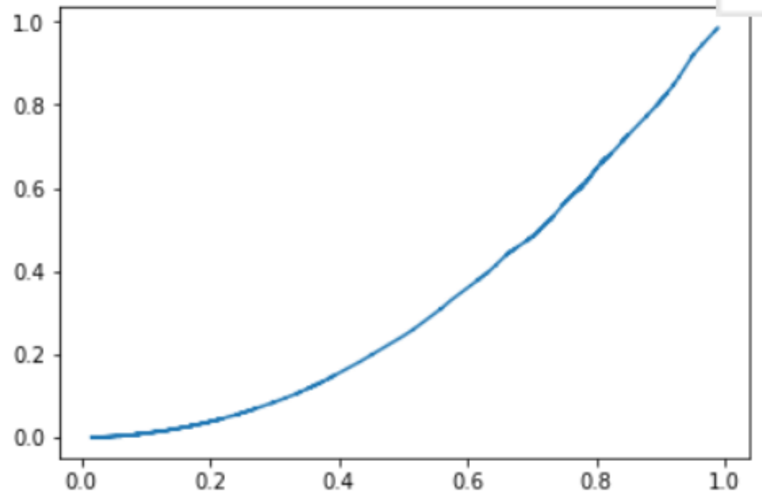
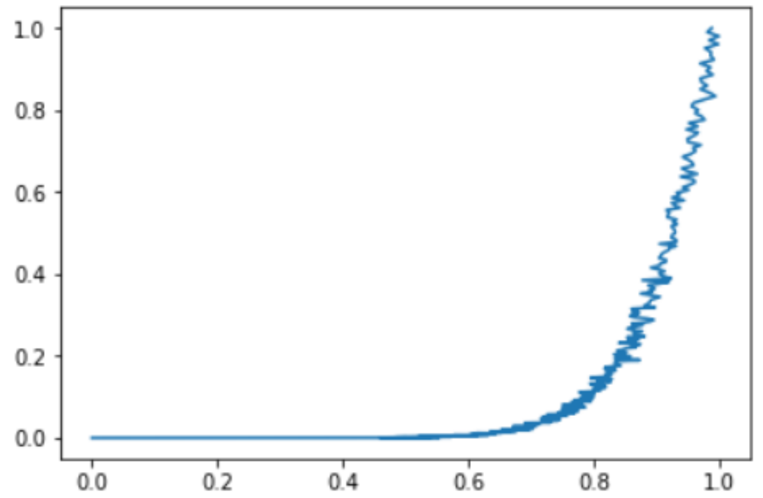
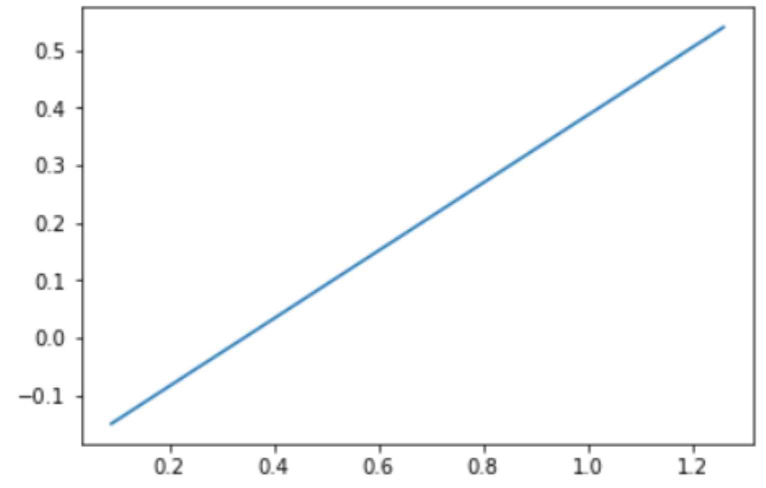
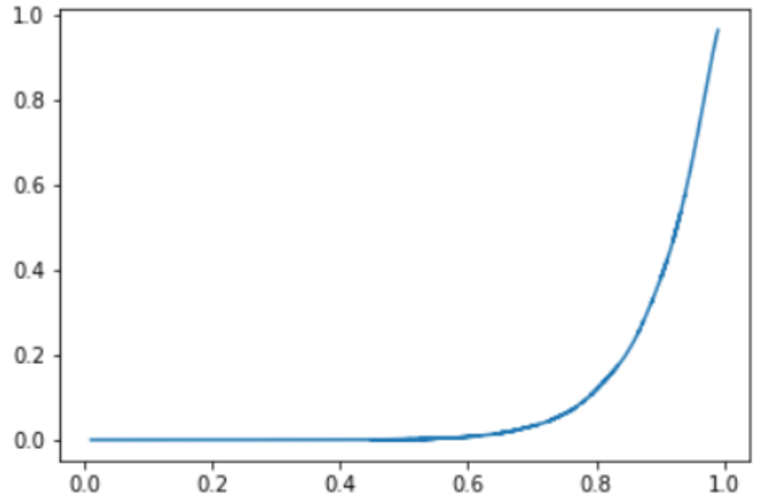
PCA vs Autoencoder

- PCA is essentially a linear transformation but autoencoders are capable of modelling complex non linear functions.
- PCA features are totally linearly uncorrelated with each other since features are projections onto the orthogonal basis. But autoencoded features might have correlations since they are just trained for accurate reconstruction.
- PCA is faster and computationally cheaper than autoencoders.
- A single layered autoencoder with a linear activation function is very similar to PCA.
- Autoencoder is prone to overfitting due to high number of parameters, but regularization and careful design can avoid this)

Credit: https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

PCA vs Autoencoder

Reconstructions on the 2D feature spaces

Function	Feature Space	PCA Reconstruction	Auto Encoder Reconstruction
y=mx+c			
$y=mx^2+c$			
$y=mx^8+c$			

Credit: https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

Exercise

Here is a two dimensional feature space $y = ax^2 + c$ **where** x **and** y **are two features with non-linear relation and c is some noise. We use an autoencoder to reconstruct this feature space. What does encoder do, and what does decoder do in this autoencoder?**

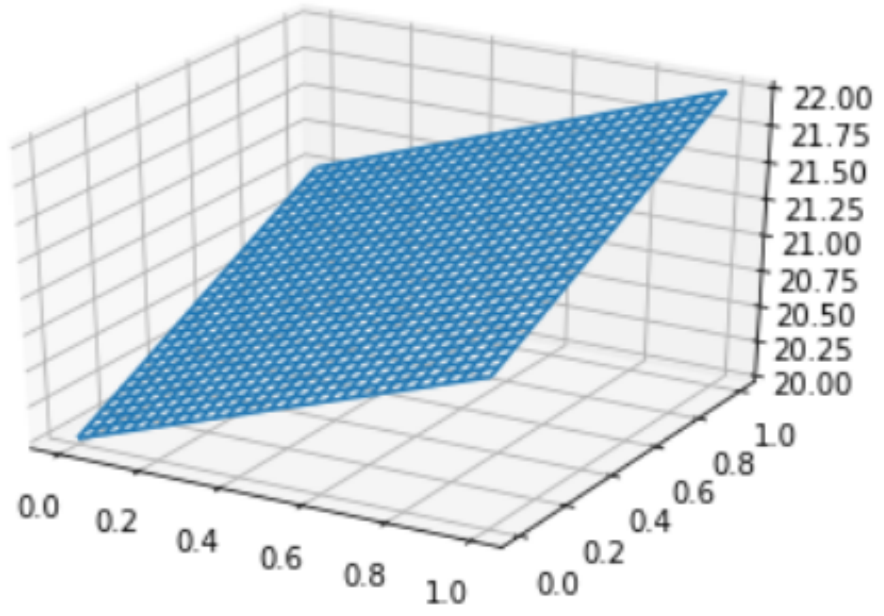
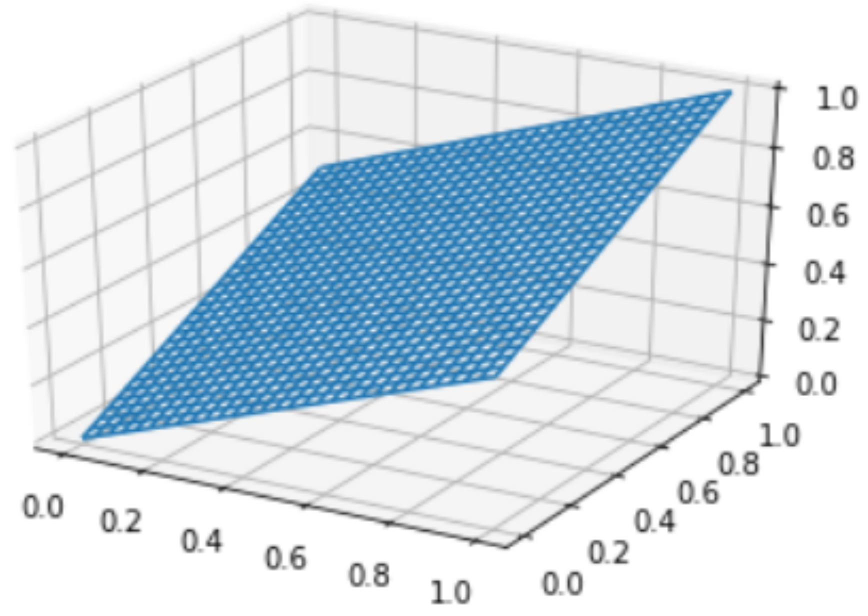
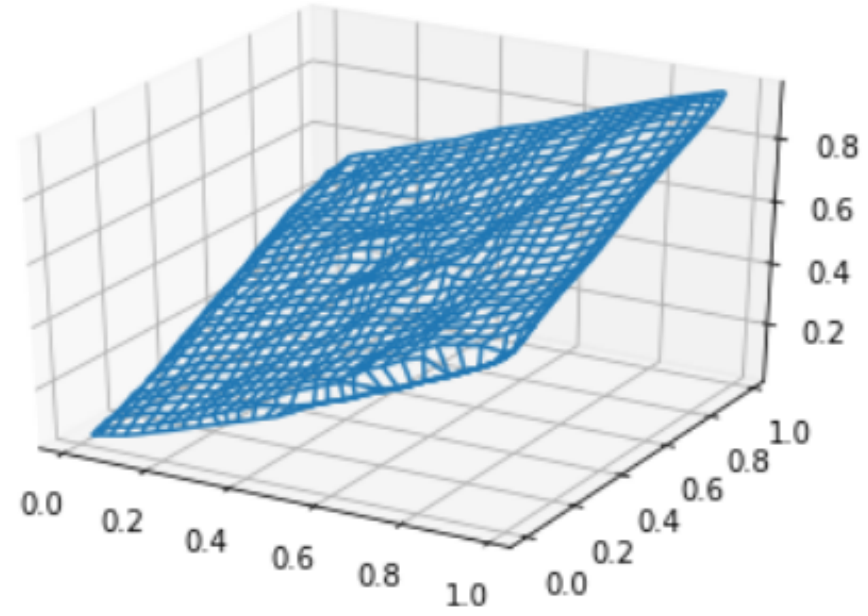
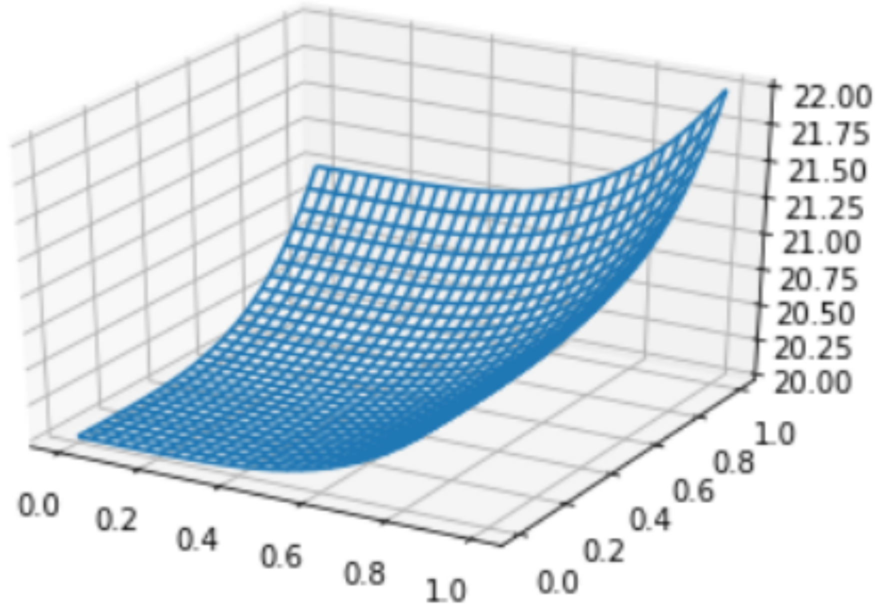
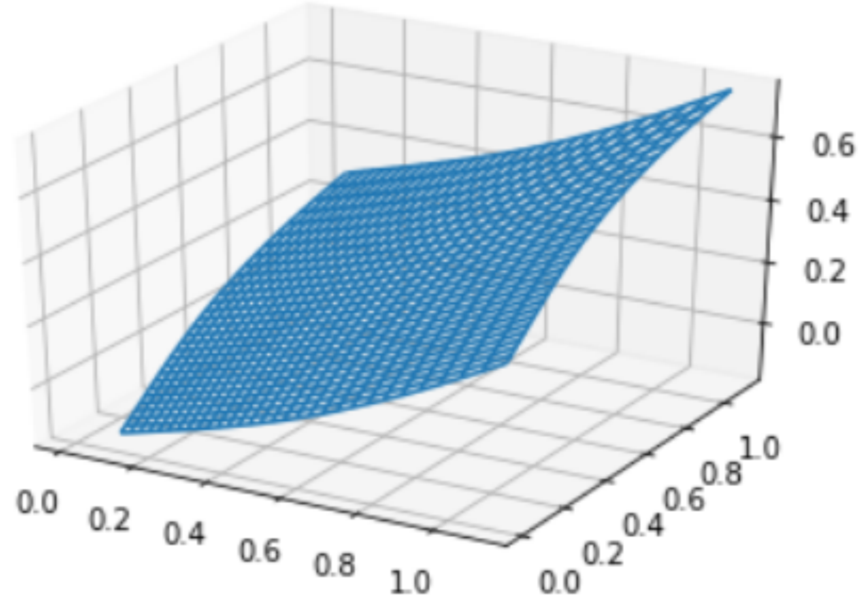
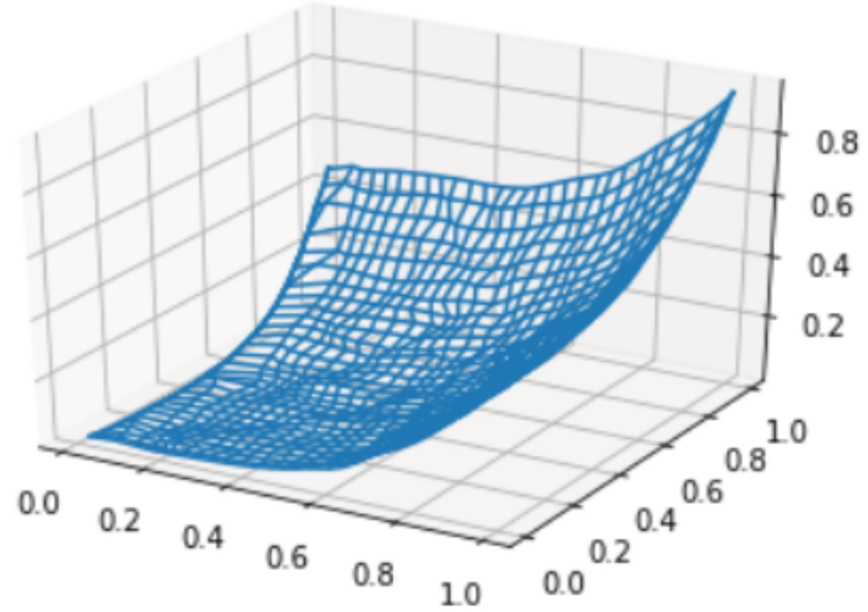
Exercise

Here is a two dimensional feature space $y = ax^2 + c$ where x and y are two features with non-linear relation and c is some noise. We use an autoencoder to reconstruct this feature space. What does encoder do, and what does decoder do in this autoencoder?

A: Encoder transforms the two dimensional input $[x, y]$ to one dimensional data $[z = ax^2]$. Then decoder reconstructs $[z]$ to two dimensional output $[\sqrt{\frac{z}{a}}, z]$.

PCA vs Autoencoder

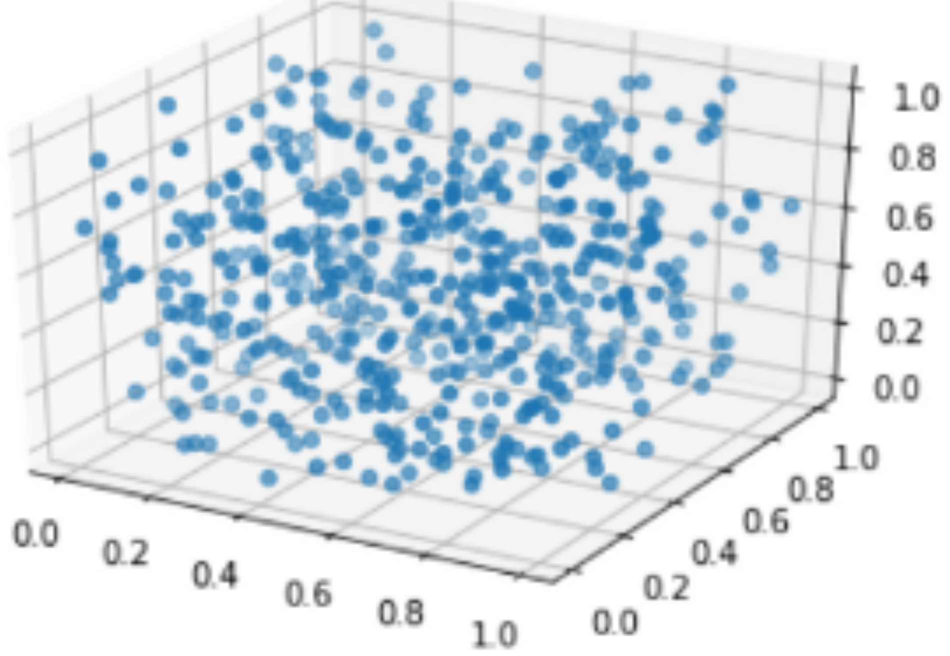
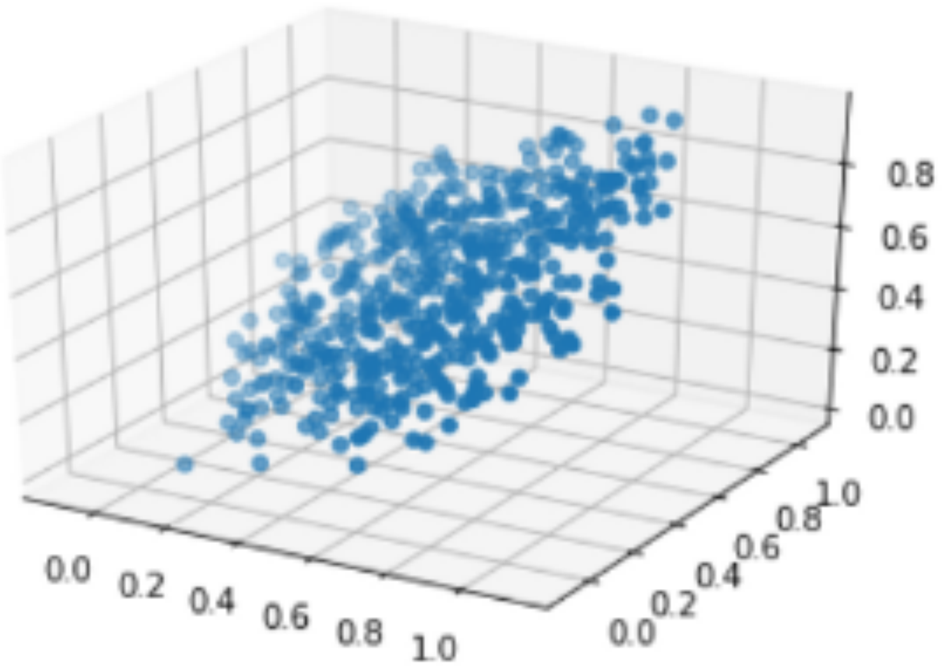
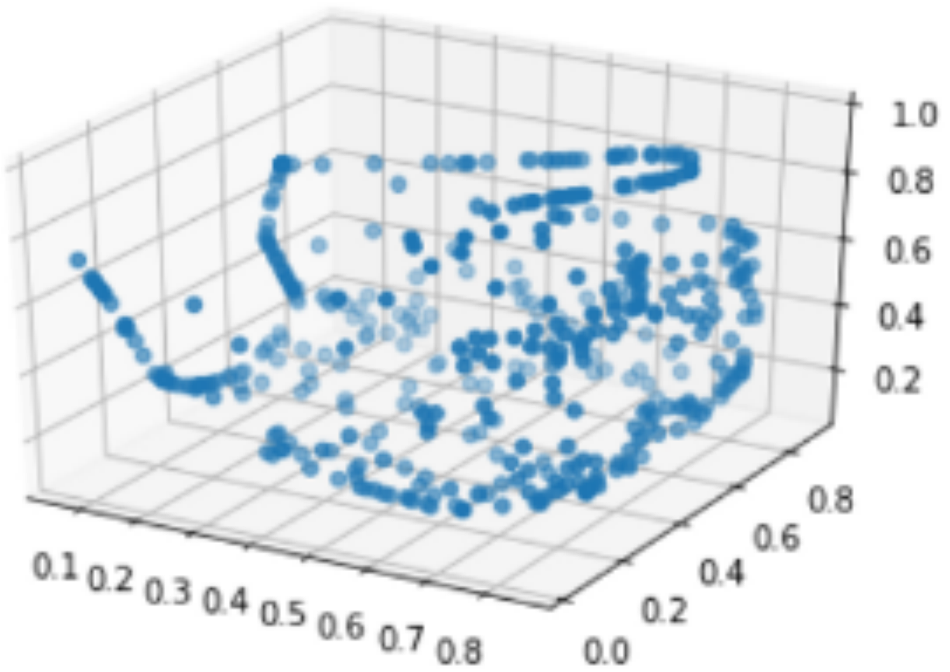
Reconstructions on the 3D feature spaces

Function	Feature Space	PCA Reconstruction	Auto Encoder Reconstruction
Plane			
Curved Surface			

Credit: https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

PCA vs Autoencoder

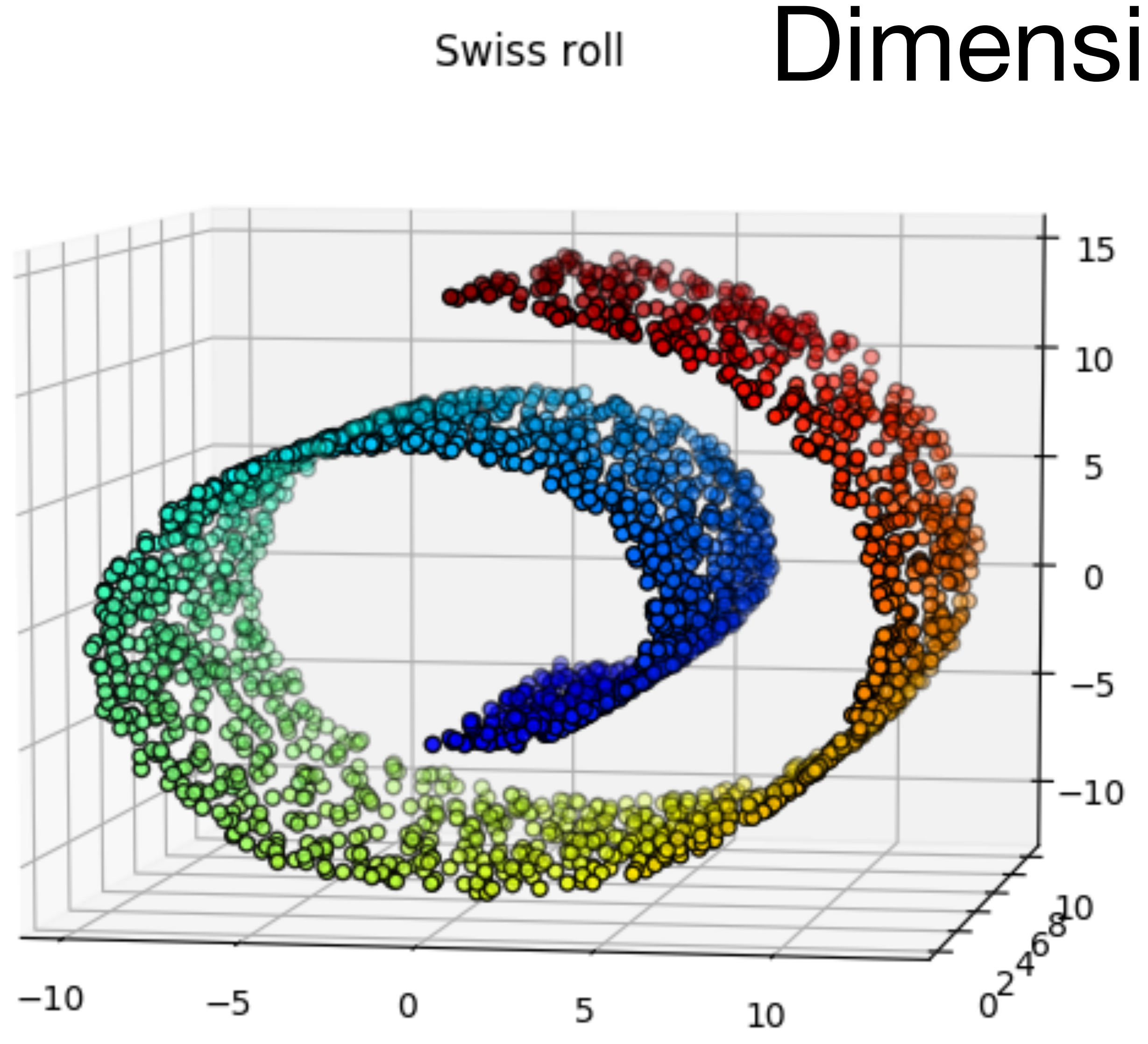
Reconstructions on the random data (without any collinearity)

	Feature Space	PCA Reconstruction	Auto Encoder Reconstruction
Random Data			
Reconstruction Cost (MSE)		0.024	0.010

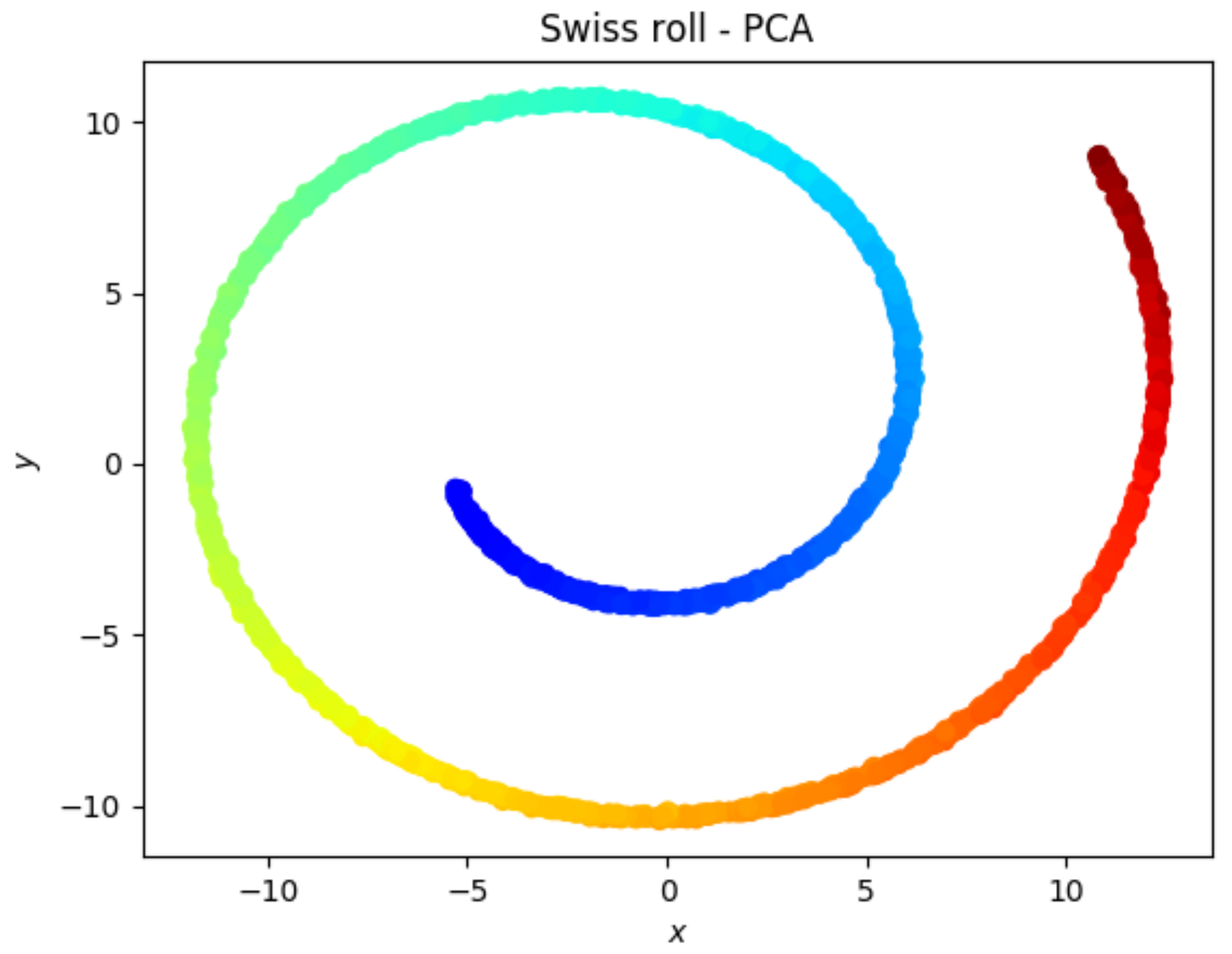
Credit: https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

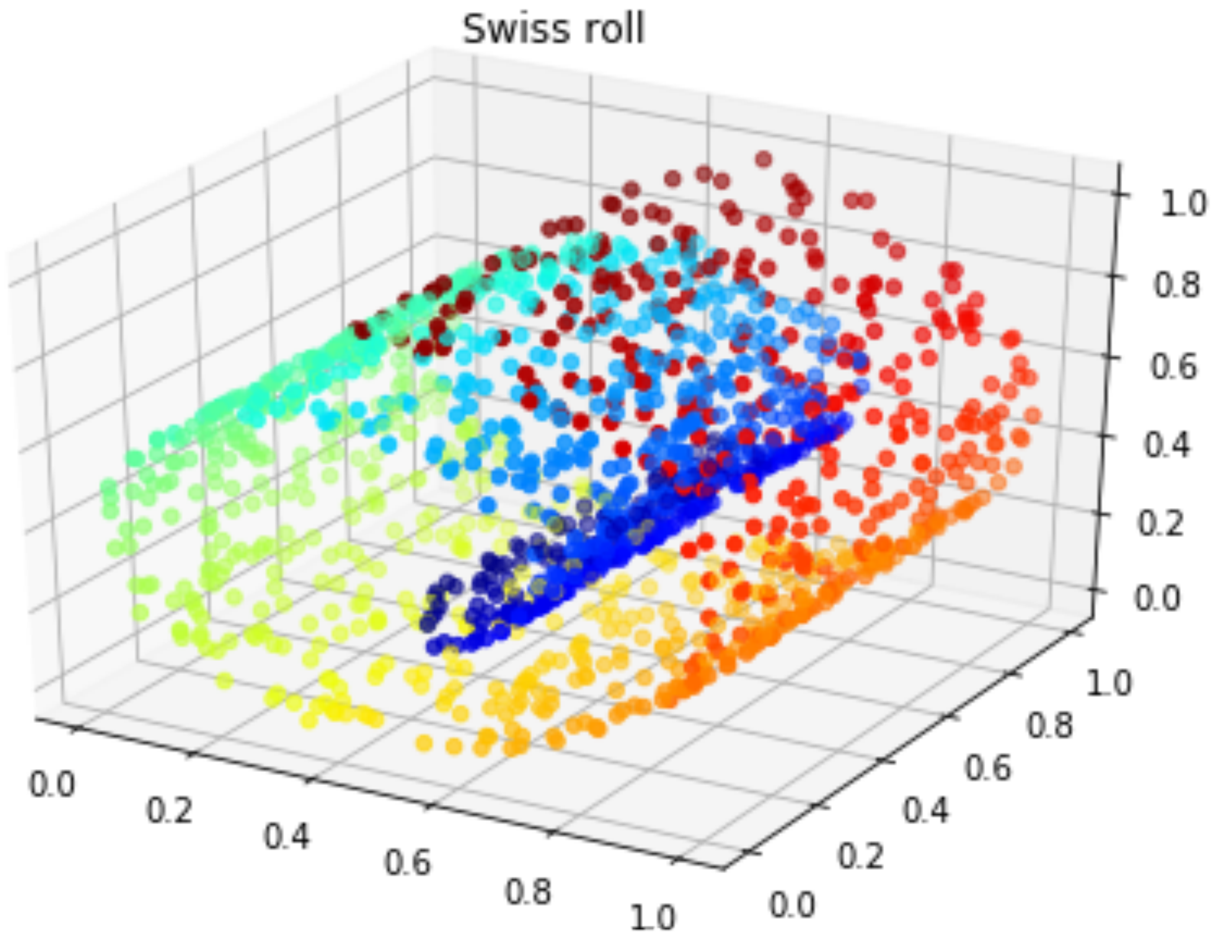
PCA vs Autoencoder

Dimensionality reduction on the Swiss roll

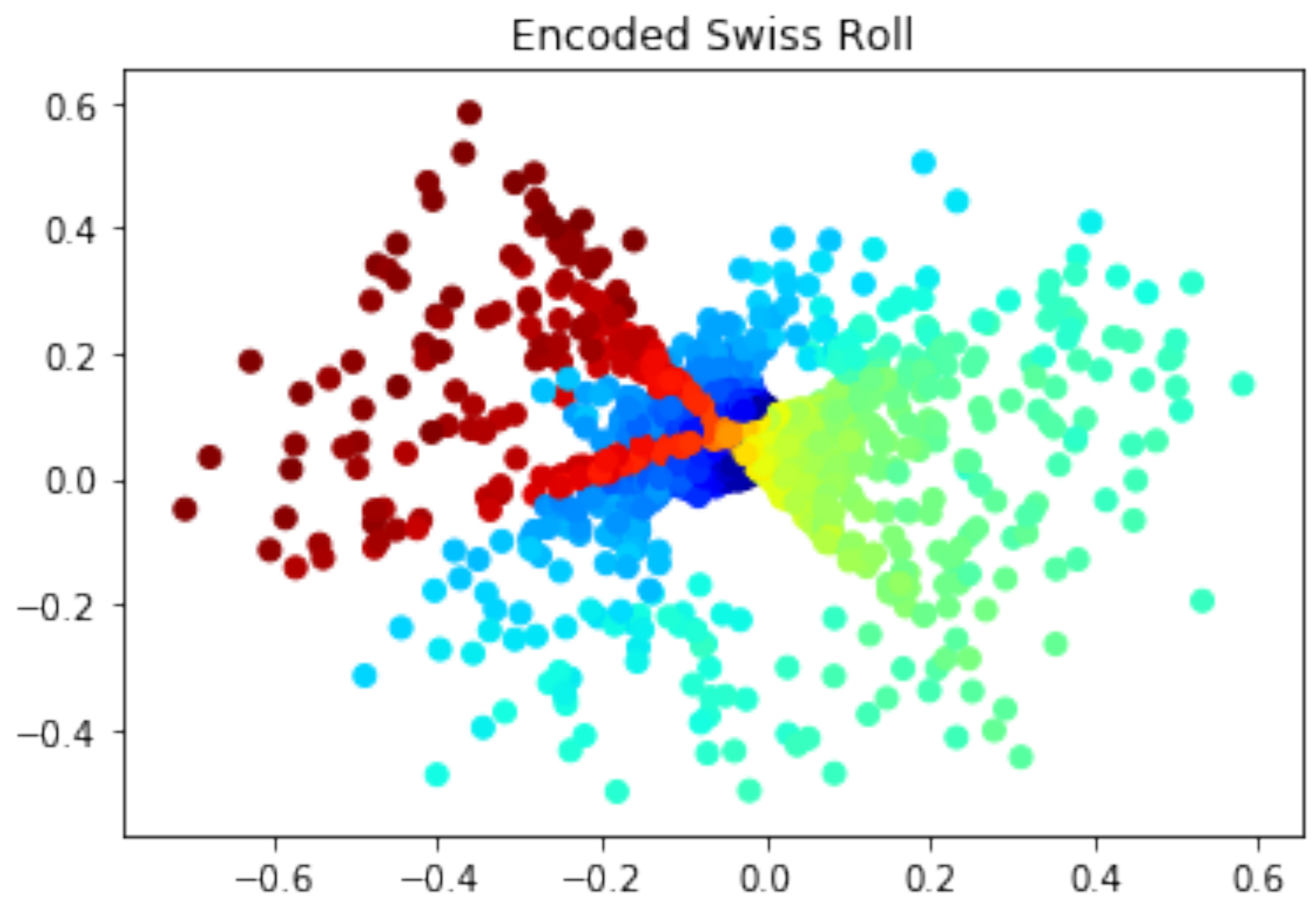


PCA





Autoencoder



PCA vs Autoencoder

Takeaway: If there is non-linearity or curvature in low dim structure, then autoencoders can encode more information using less dimensions.

Credit: https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

Structure of Autoencoder

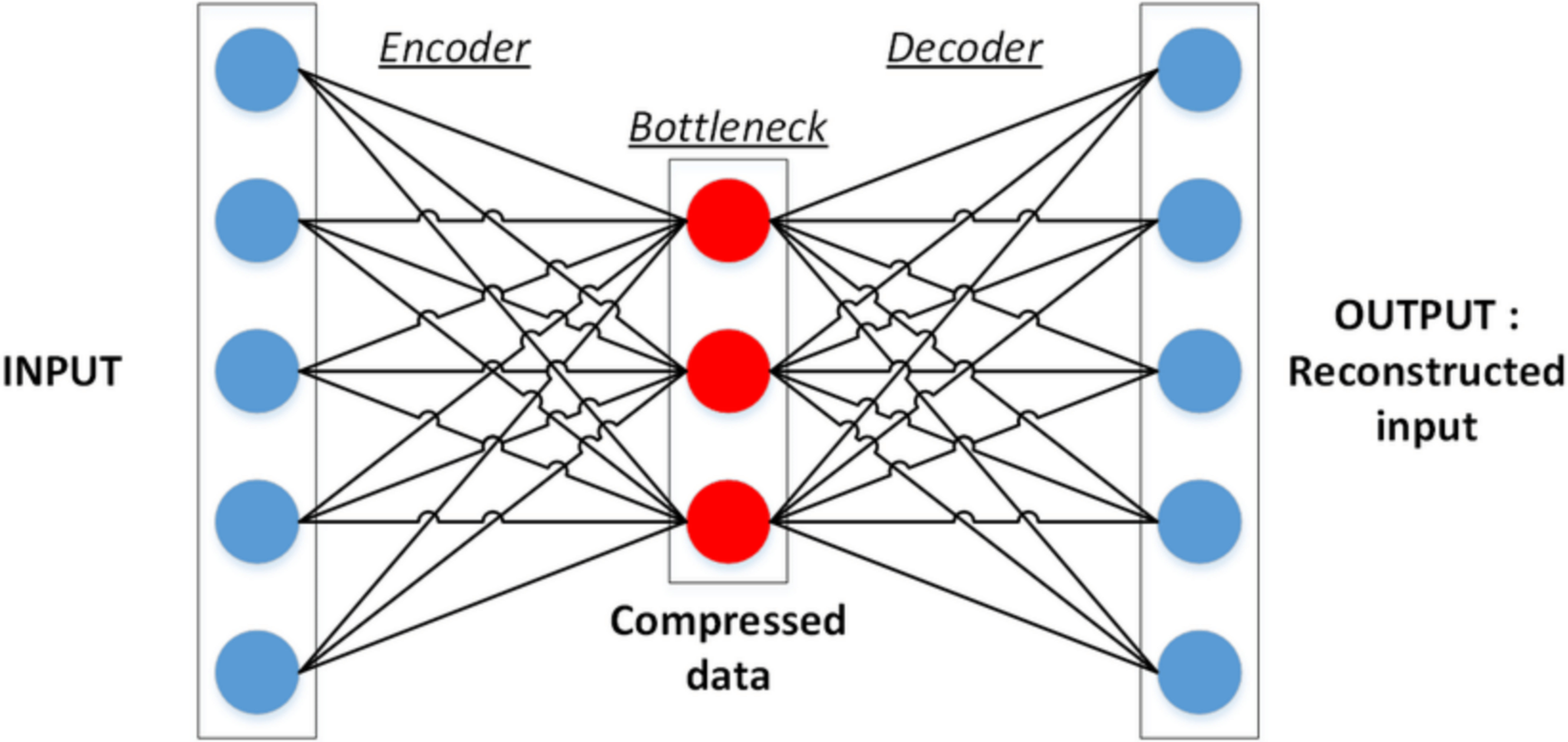
Encoder: In which the model learns how to reduce the input dimensions and compress the input data into an encoded representation.

Bottleneck: which is the layer that contains the compressed representation of the input data. This is the lowest possible dimensions of the input data.

Decoder: In which the model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.

Credit: https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726

Structure of Autoencoder



Credit: https://stackabuse.com/autoencoders-for-image-reconstruction-in-python-and-keras/

Applications of Autoencoder

- Dimensionality Reduction

- Image Denoising

- Feature Extraction

  Encoding part of Autoencoders helps to learn important hidden features present in the input data, in the process to reduce the reconstruction error.

- Image Generation

  There is a type of Autoencoder, named Variational Autoencoder(VAE), this type of autoencoders are ***Generative Model***, used to generate images.

  The idea is that given input images like images of face or scenery, the system will generate similar images.

Credit: https://iq.opengenus.org/applications-of-autoencoders/