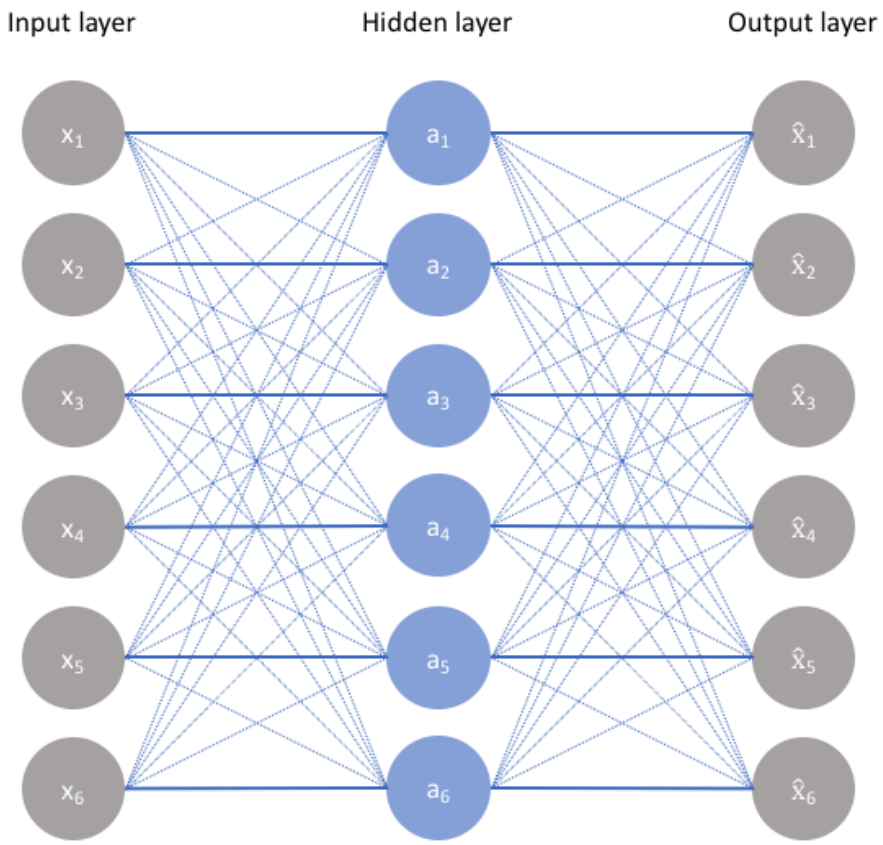
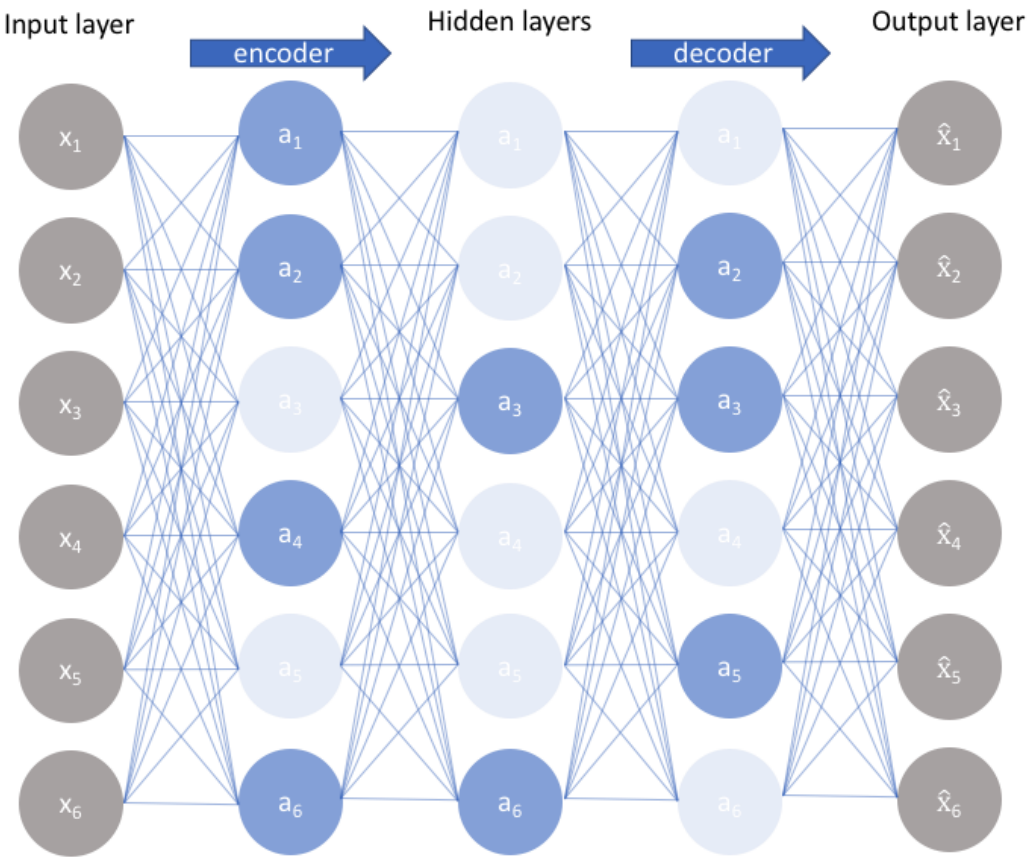
Why do autoencoders have fewer neurons in the hidden layer than in the input & output layer, and what happens if we initialize the hidden layer with equal (or more) neurons than there are neurons in the input & output layer?

We design a neural network architecture such that we impose a bottleneck in the network which forces a **compressed** knowledge representation of the original input.
Our only way to ensure that the model isn't memorizing the input data is to ensure that we've sufficiently restricted the number of nodes in the hidden layer(s).
Without the presence of an information bottleneck, our network could easily learn to simply memorize the input values by passing these values along through the network (visualized below).



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. Rather, we'll construct our loss function such that we penalize *activations* within a layer.



Whereas an undercomplete autoencoder will use the entire network for every observation, a sparse autoencoder will be forced to selectively activate regions of the network depending on the input data. As a result, we've limited the network's capacity to memorize the input data without limiting the networks capability to extract features from the data.

Overcomplete Autoencoders are a type of autoencoders that use more hidden units than the number of input units. This means that the encoder and decoder layers have more units than the input layer. The idea behind using more hidden units is to learn a more complex, non-linear function that can better capture the structure of the data.

Advantages of using Overcomplete Autoencoders include their ability to learn more complex representations of the data, which can be useful for tasks such as feature extraction and denoising. Overcomplete Autoencoders are also more robust to noise and can handle missing data better than traditional autoencoders

However, there are also some disadvantages to using Overcomplete Autoencoders. One of the main disadvantages is that they can be more difficult to train than traditional autoencoders. This is because the extra hidden units can cause overfitting, which can lead to poor performance on new data.