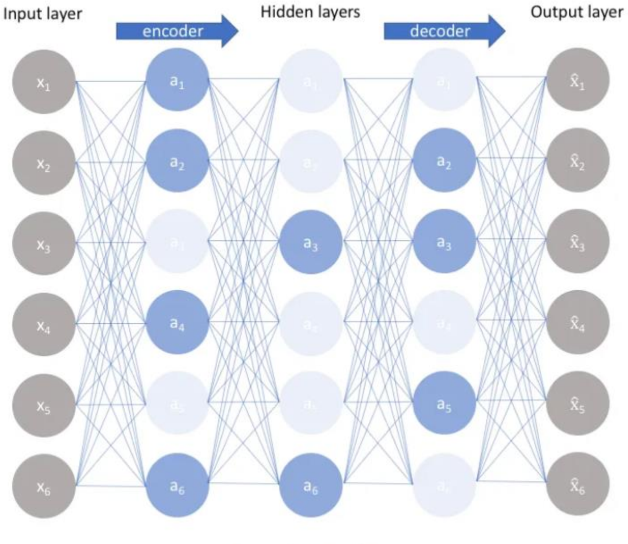
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Sparse Autoencoders

A sparse autoencoder is simply an autoencoder whose training criterion involves a sparsity penalty. In most cases, we would construct our loss function by penalizing activations of hidden layers so that only a few nodes are encouraged to activate when a single sample is fed into the network.

The intuition behind this method is that, for example, if a man claims to be an expert in mathematics, computer science, psychology, and classical music, he might be just learning some quite shallow knowledge in these subjects. However, if he only claims to be devoted to mathematics, we would like to anticipate some useful insights from him. And it's the same for autoencoders we're training — fewer nodes activating while still keeping its performance would guarantee that the autoencoder is actually learning latent representations instead of redundant information in our input data.



Sparse Autoencoder

There are actually two different ways to construct our sparsity penalty: L1 regularization and KL-divergence. Here we will only talk about L1 regularization.

Although L1 and L2 can both be used as regularization term, the key difference between them is that L1 regularization tends to shrink the penalty coefficient to zero while L2 regularization would move coefficients towards zero but they will never reach. Thus L1 regularization is often used as a method of feature extraction.

$$L_1 = \|w\|\ , L_2 = w^2$$

The idea of using L1 regularization in sparse autoencoder and the loss function is as below:

$$Obj = L(x, \hat{x}) + regularization + \lambda \sum_i |a_i^{(h)}|$$

Except for the first two terms, we add the third term which penalizes the absolute value of the vector of activations a in layer h for sample i. Then we use a hyperparameter to control its effect on the whole loss function. And in this way, we do build a sparse autoencoder.

Due to the sparsity of L1 regularization, sparse autoencoder actually learns better representations and its activations are more sparse which makes it perform better than original autoencoder without L1 regularization.