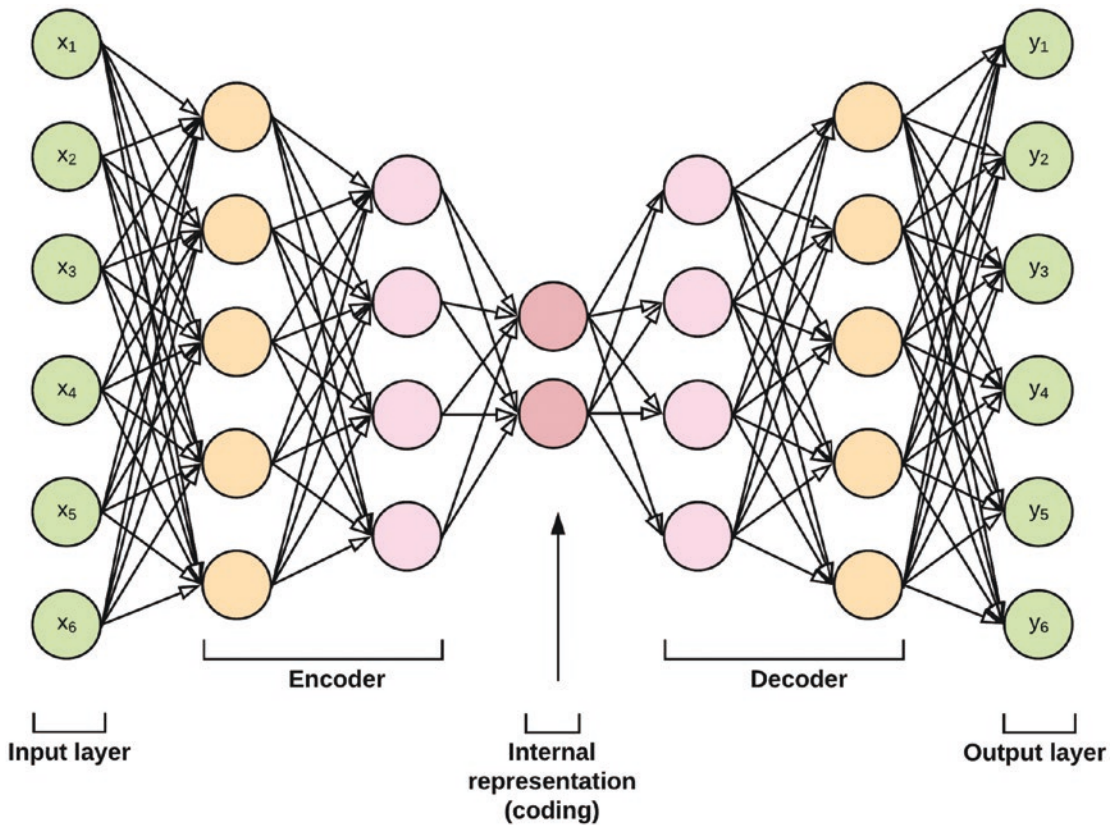


Figure 37-1. A simple autoencoder architecture

## Stacked Autoencoders

Stacked autoencoder is when the simple autoencoder architecture as shown in Figure 37-1 is enhanced with multiple hidden layers. Just like other deep neural network architectures with hidden layers, the hidden layers of an autoencoder enable the network to learn more complex patterns of the input dataset.

The hidden layers of a stacked or deep autoencoder are added symmetrically at both the Encoder and Decoder part of the network as shown in Figure 22-2. The neurons of the hidden layers are restricted to be less than that of the input layer. This formulation places a restriction on the network, so it doesn't merely memorize the input. Moreover, care must be taken not to create too many deep layers, so the autoencoder does not overfit the input data and fail to generalize to out-of-sample examples. To optimize the training of a deep autoencoder, the weights of the symmetrical neural layers are shared in a technique called *tying*.



**Figure 37-2.** Stacked or deep autoencoder. The hidden layers are added symmetrically at both the Encoder and Decoder

## Stacked Autoencoders with TensorFlow 2.0

The code example in this section shows how to implement an autoencoder network using TensorFlow 2.0. For simplicity, the MNIST handwriting dataset is used to create reconstructions of the original images. In this example, a stacked autoencoder is implemented with the original and reconstructed image shown in Figure 37-3. The code listing is presented in the following, and corresponding notes on the code are shown thereafter.

```
# import TensorFlow 2.0 with GPU
!pip install -q tf-nightly-gpu-2.0-preview

# import packages
import tensorflow as tf
```