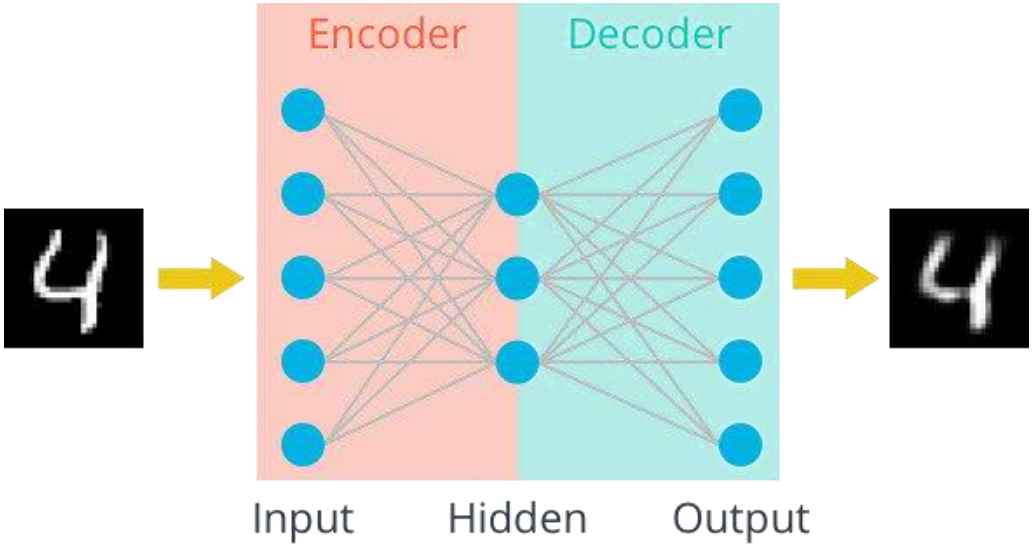
Module 3

Linear Autoencoder

We'll train an autoencoder with these images by flattening them into 784 length vectors. The images from this dataset are already normalized such that the values are between 0 and 1. Let's start by building a simple autoencoder. The encoder and decoder should be made of **one linear layer**. The units that connect the encoder and decoder will be the *compressed representation*.

Since the images are normalized between 0 and 1, we need to use a **sigmoid activation on the output layer** to get values that match this input value range.



An autoencoder is a neural network that can be used to encode and decode data. The general structure of an autoencoder is shown in the figure below. It consists of two parts: an encoder and a decoder. The encoder compresses the input data into a lower dimensional representation ("code" in the schematic below, which is often referred to as the *latent space representation*) by extracting the most salient features of the data, while the decoder reconstructs the input data from the compressed representation. Therefore, autoencoder is often used for *dimensionality reduction*. In this tutorial, our goal is to compare the performance of two types of autoencoders, a linear autoencoder and a convolutional autoencoder, on reconstructing the Fashion-MNIST images. With the help of Covalent, we will see how to break a complex workflow into smaller and more manageable tasks, which allows users to track the task dependencies and execution results of individual steps. Another advantage of Covalent is its ability to auto-parallelize the execution of subtasks.