Building Unsupervised Machine Learning Models

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Unsupervised machine learning models

Autoencoders to find latent features in data

Build and train an autoencoder using Keras layers

Types of ML Algorithms

Supervised

Labels associated with the training data is used to correct the algorithm

Unsupervised

The model has to be set up right to learn structure in the data

Unsupervised Learning

Only have input data x – no output data

Model the underlying structure to learn more about data

Algorithms self discover the patterns and structure in the data

Unsupervised ML Algorithms

Clustering

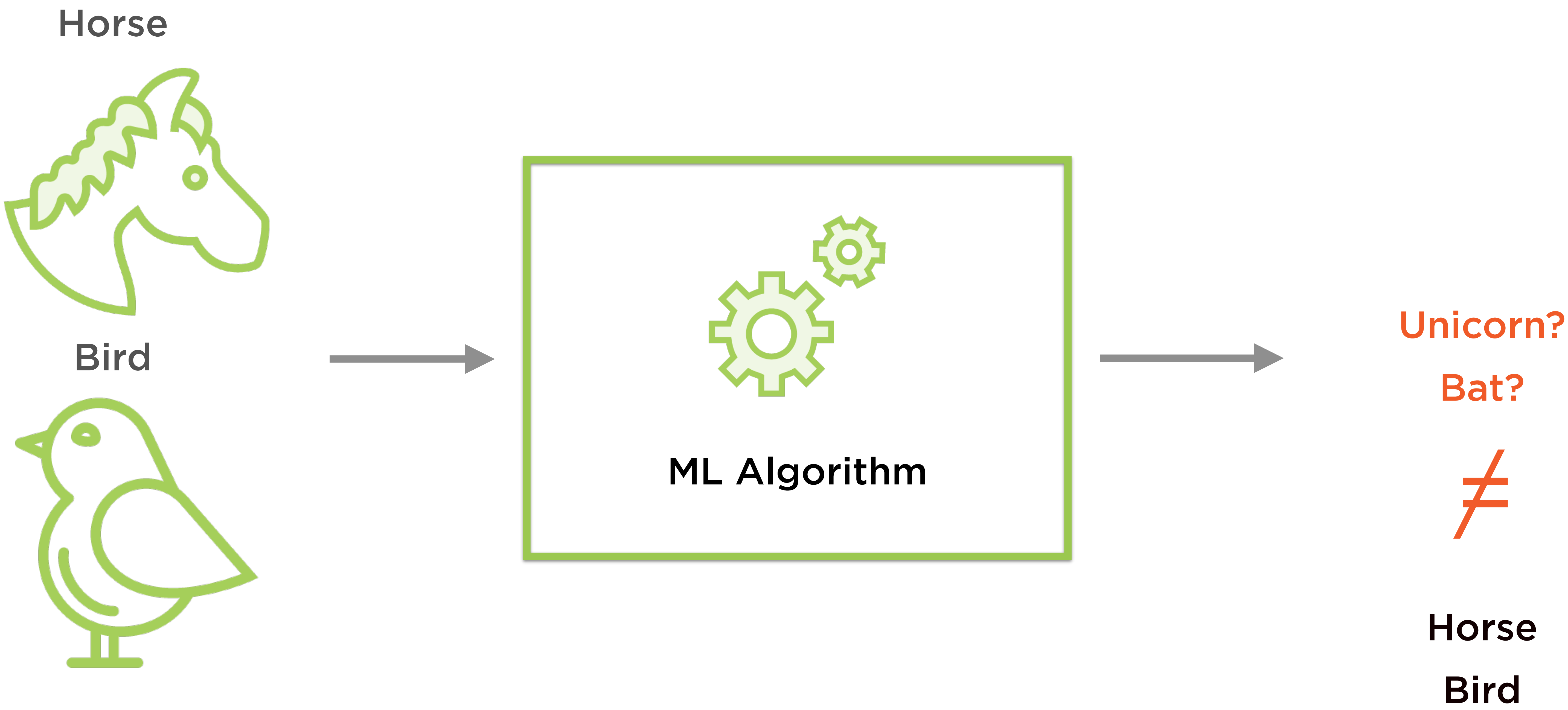
Identify patterns in data items e.g. K-means clustering

Autoencoding

Identify latent factors that drive data e.g. PCA

Autoencoders for Efficient Representation

Supervised Machine Learning



Horse

Bird

ML Algorithm

Unicorn?

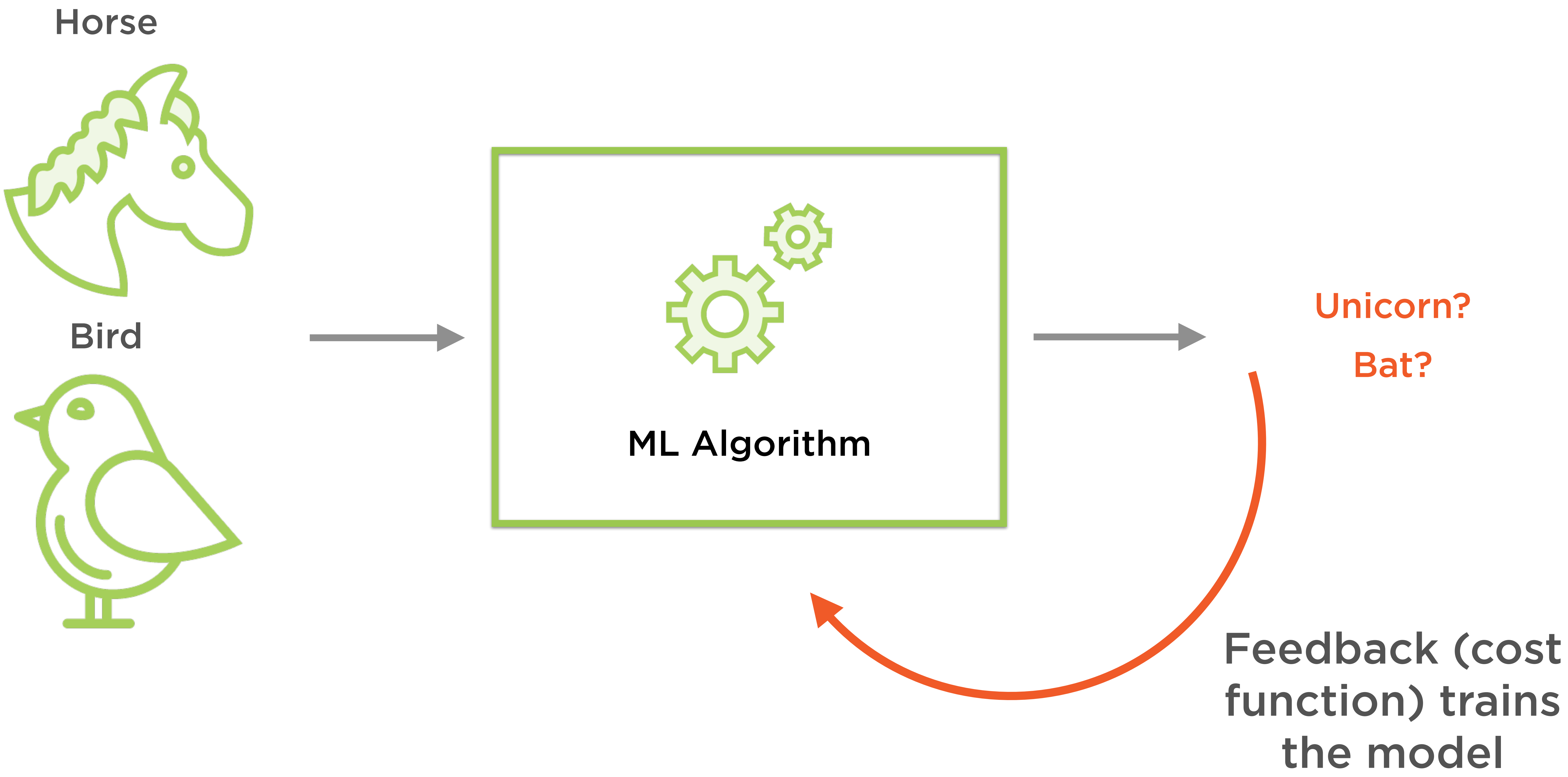
Bat?

≠

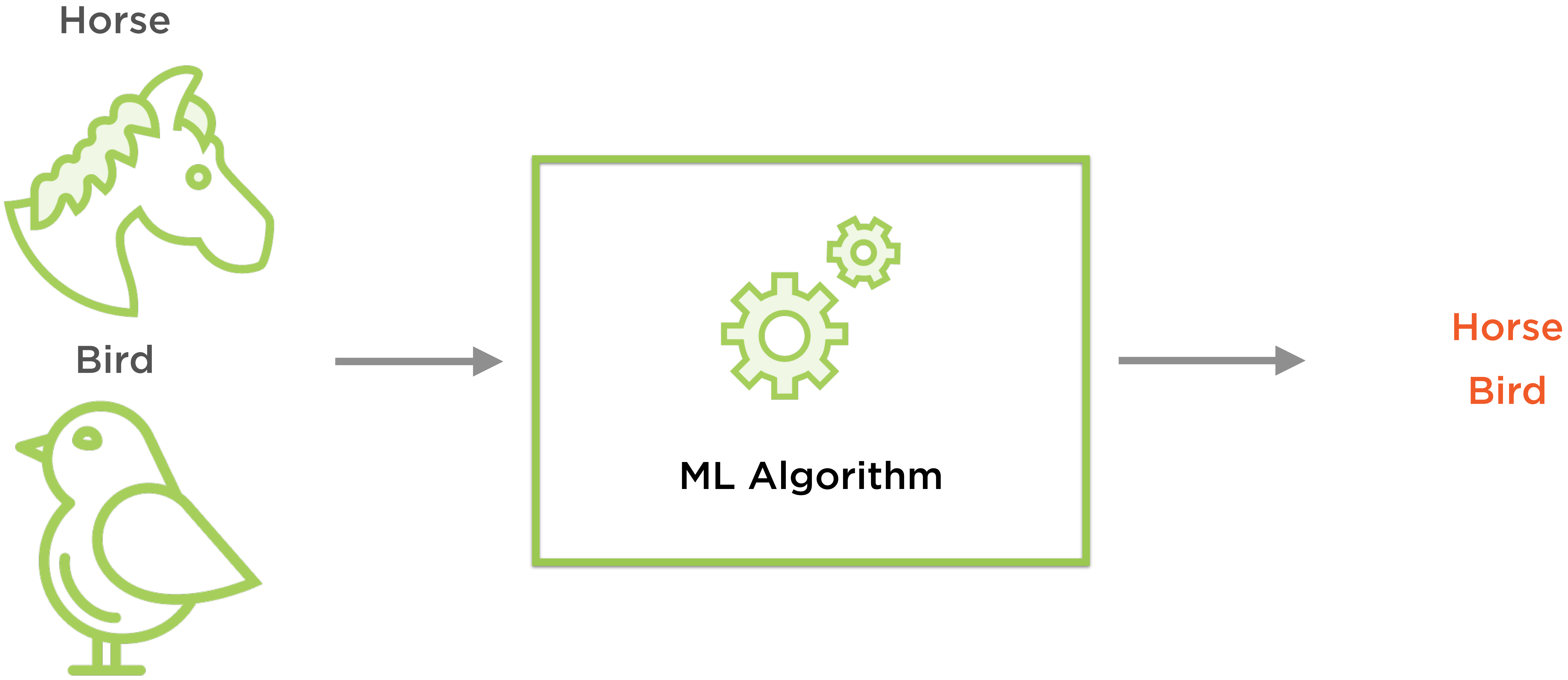
Horse

Bird

Supervised Machine Learning



Supervised Machine Learning



Horse

Bird

ML Algorithm

Horse

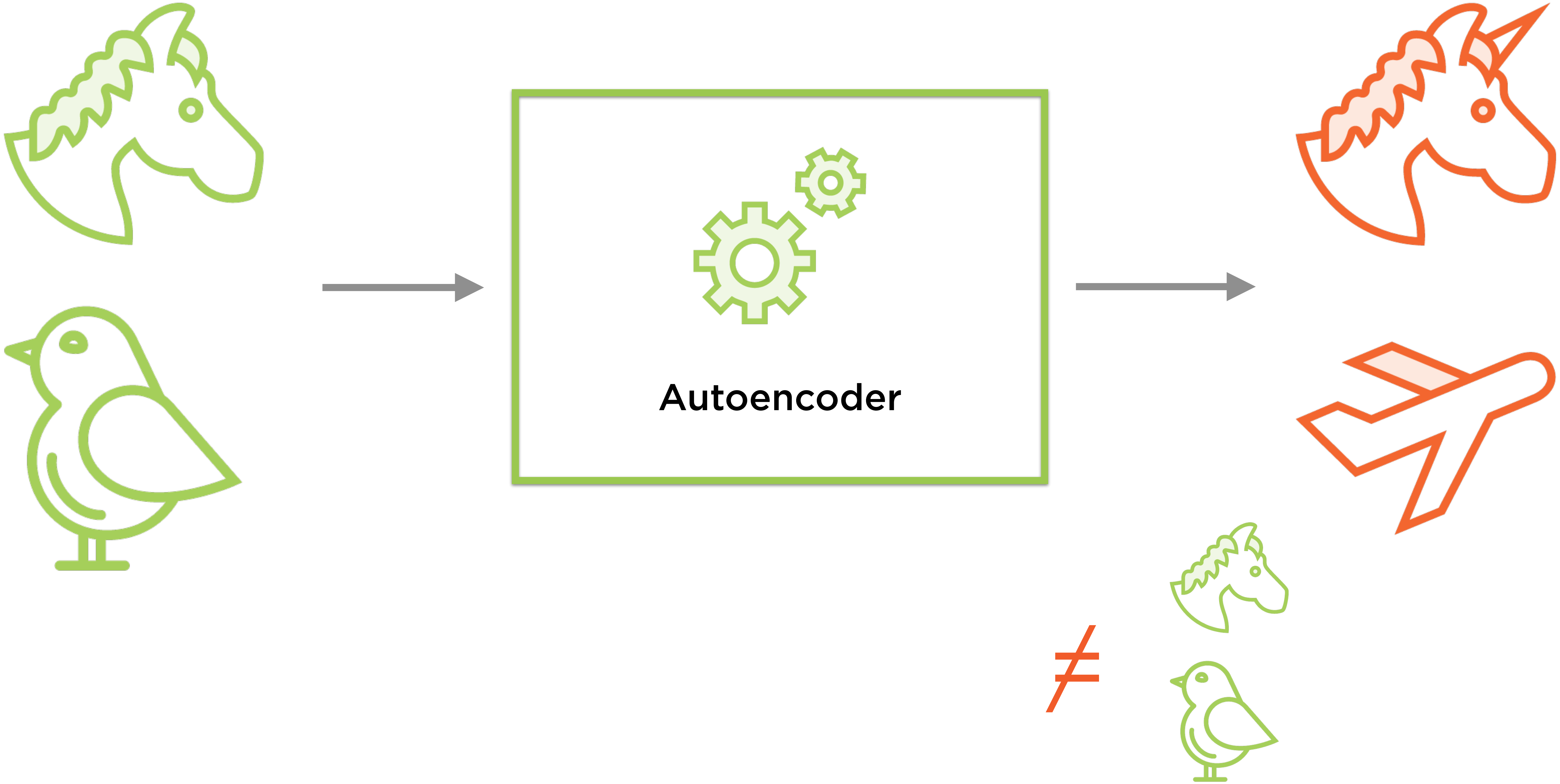
Bird

$$x = f(x)$$

Autoencoders Learn the Input!

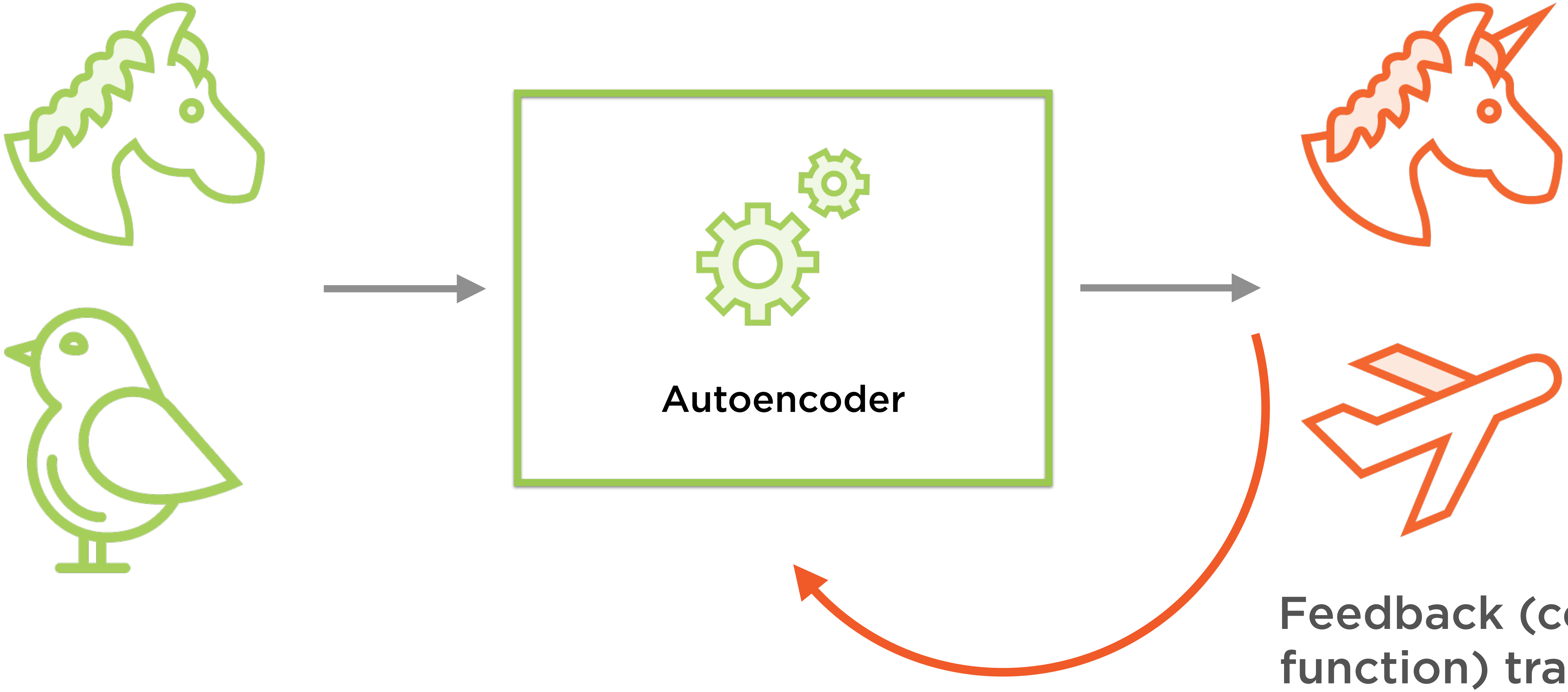
The process is inherently unsupervised, but cleverly uses the input itself to train an algorithm

Autoencoder



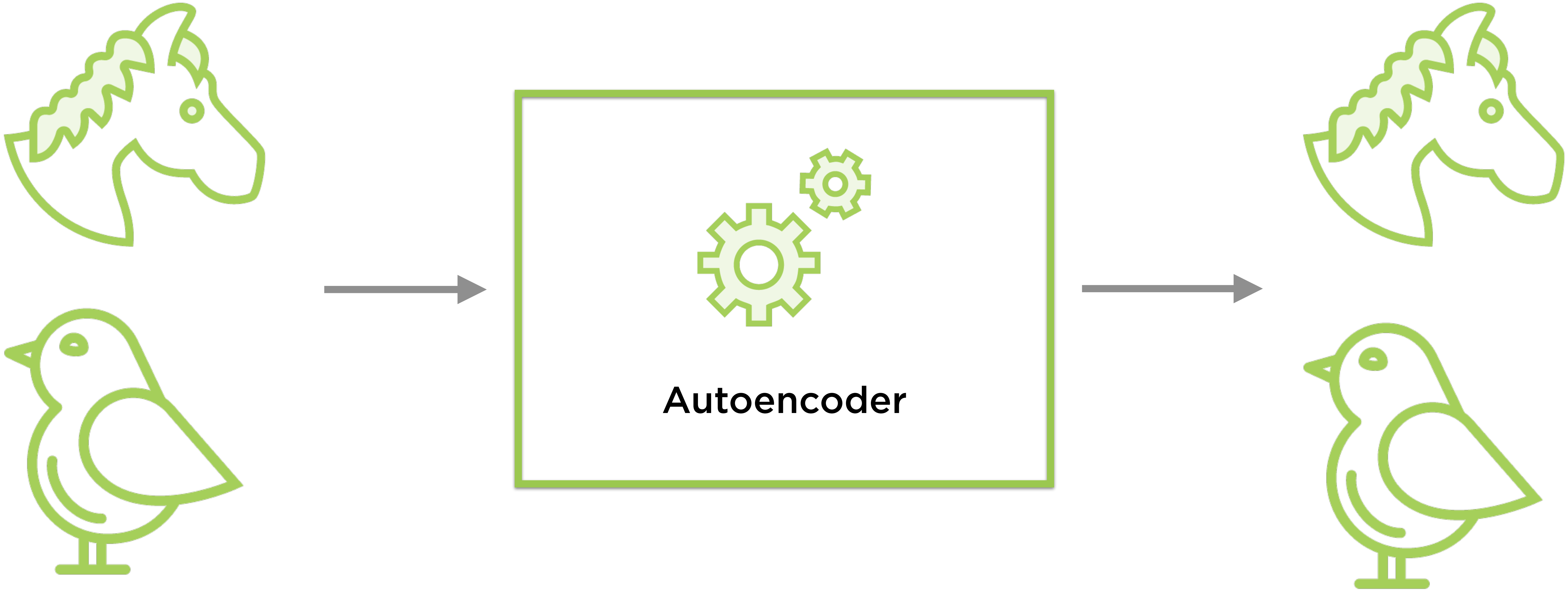
Autoencoder

Autoencoder



Feedback (cost function) trains the model

Autoencoder



$$x = f(x) = g(L)$$

Uncover Hidden Patterns in Data

The function f is just the identity, not very interesting. Autoencoders uncover latent factors L that actually drive our data

Autoencoders are Neural Networks that learn efficient representations of data (e.g. PCA)

Autoencoders

Autoencoders are the ultimate “look-within” unsupervised ML technique

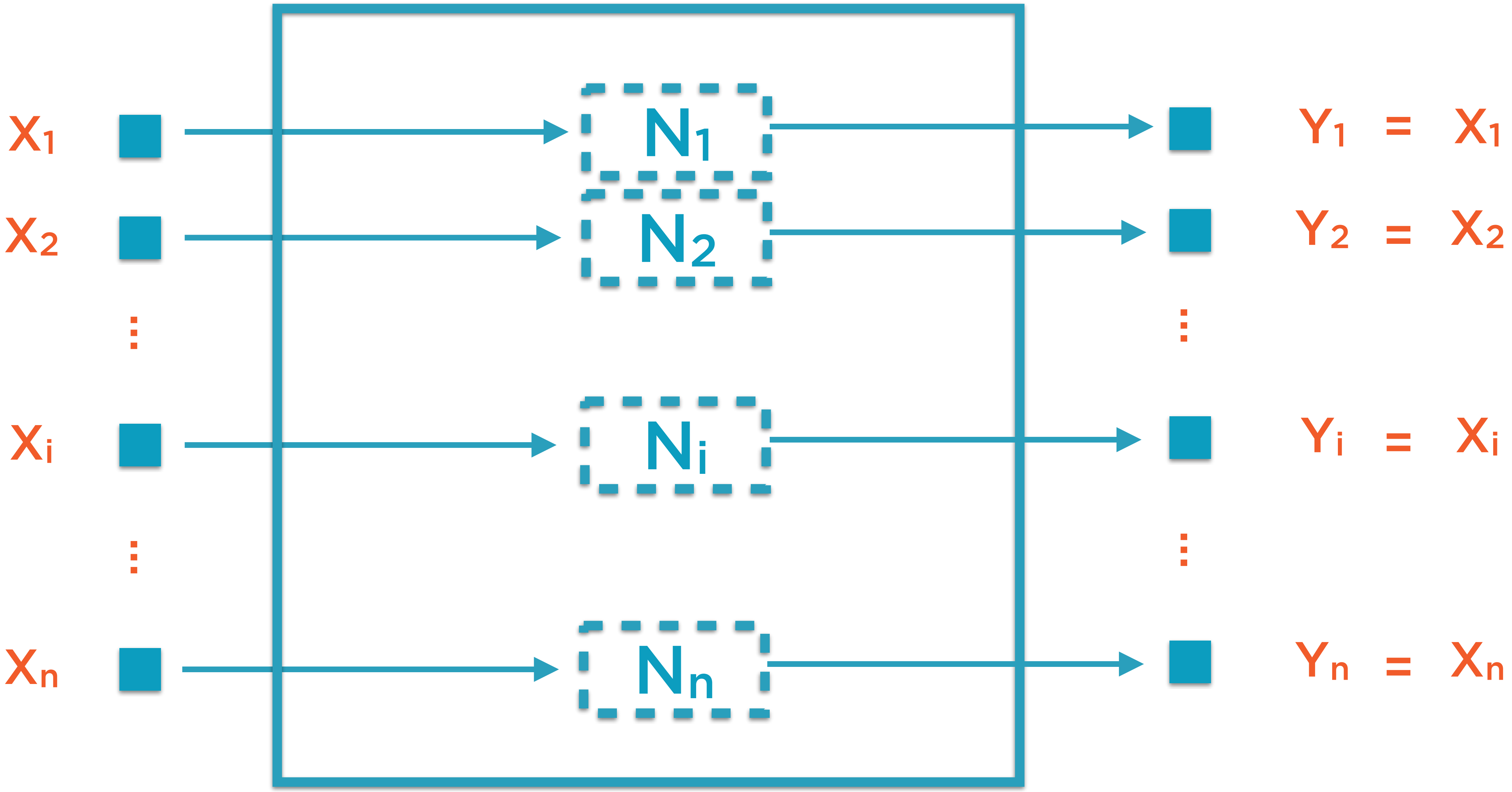
Simply neural networks that try to output exactly what is input

Autoencoders

Sounds trivial…

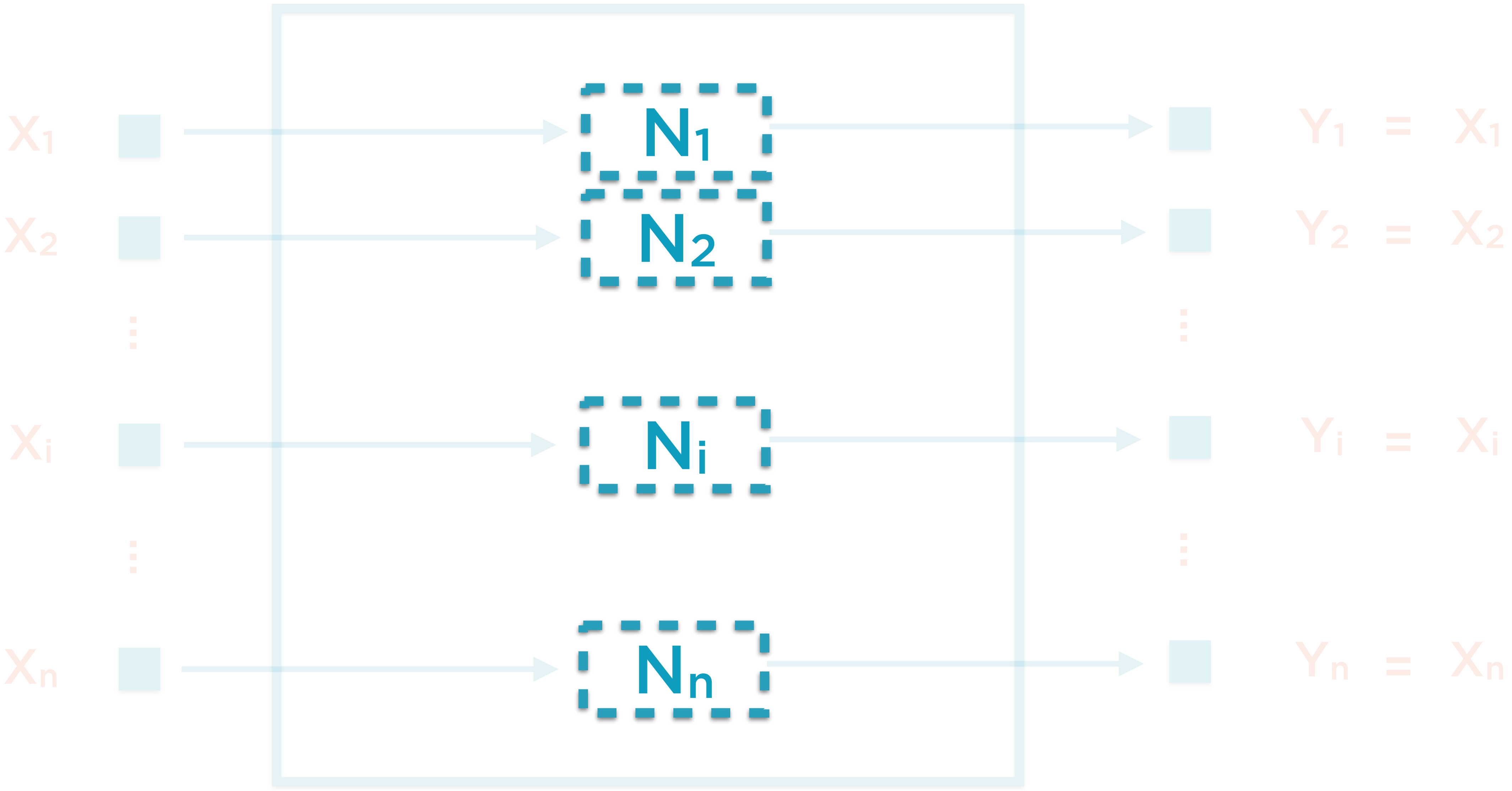
…But we constrain the NN architecture to force real learning

Trivial Autoencoder



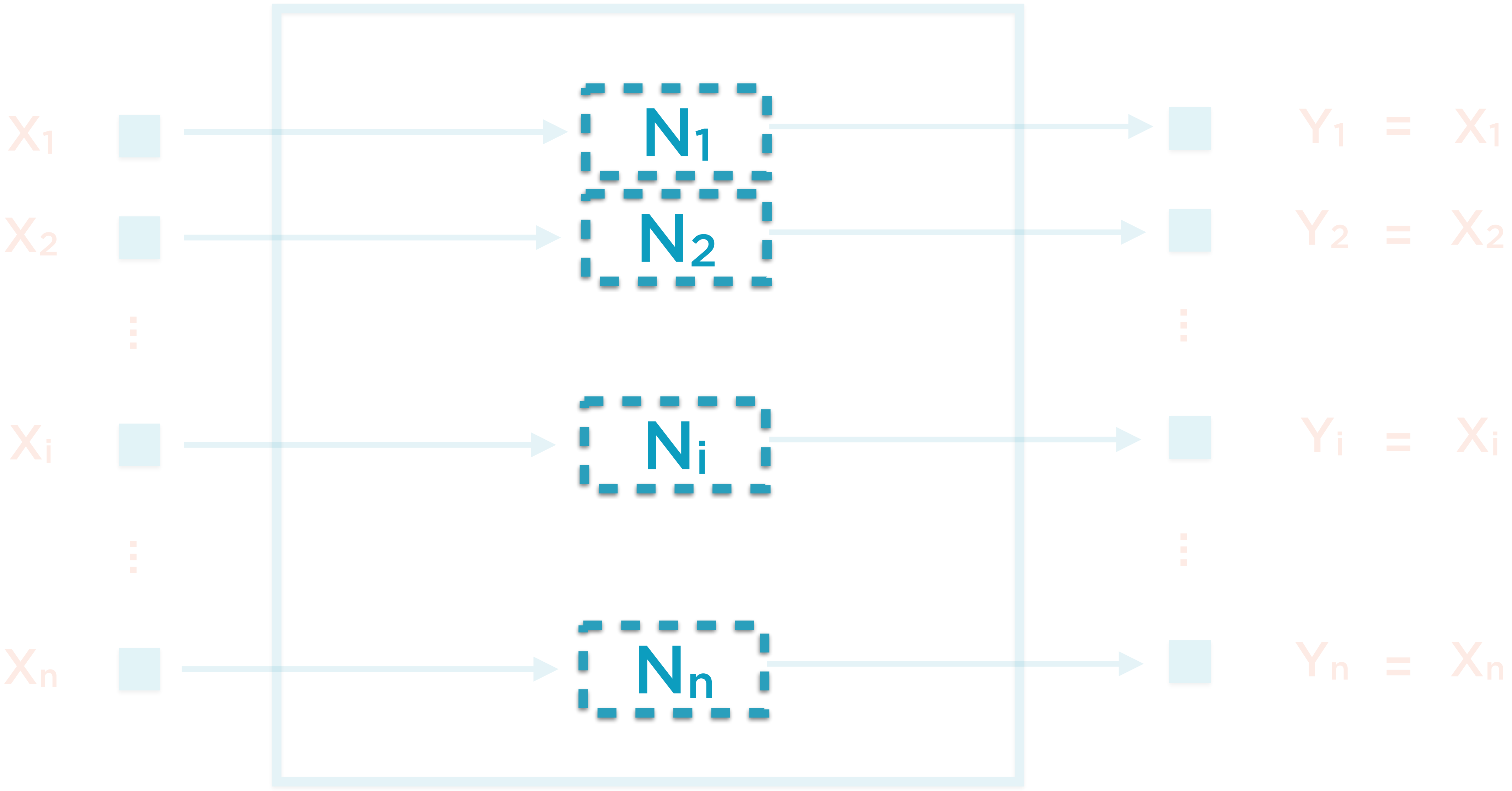
This neural network trivially “learns” the input

Trivial Autoencoder



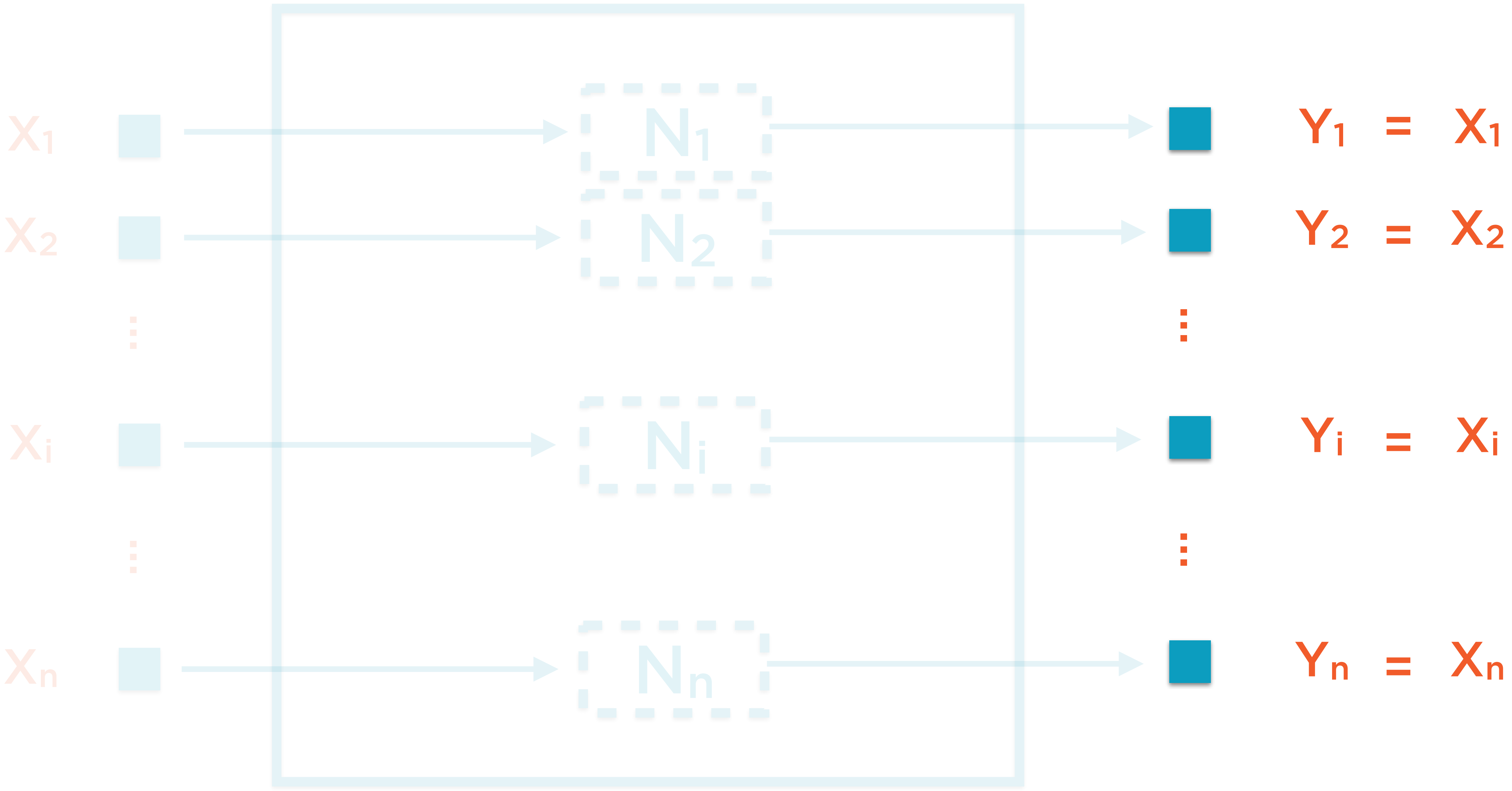
Just one layer, which serves as both input and output layer

Trivial Autoencoder



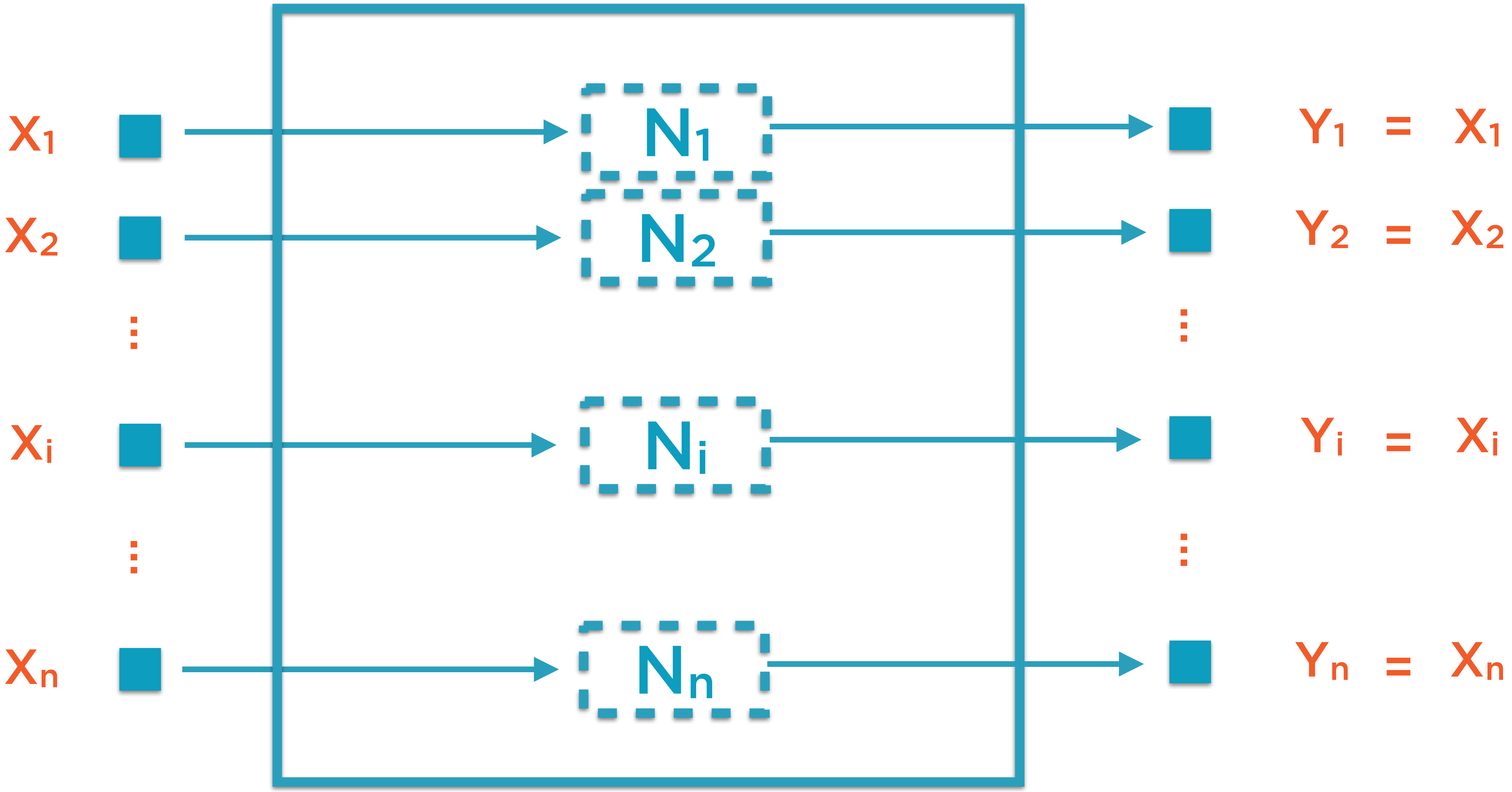
In an autoencoder, the input and output layer must have same dimensionality as the input data

Trivial Autoencoder



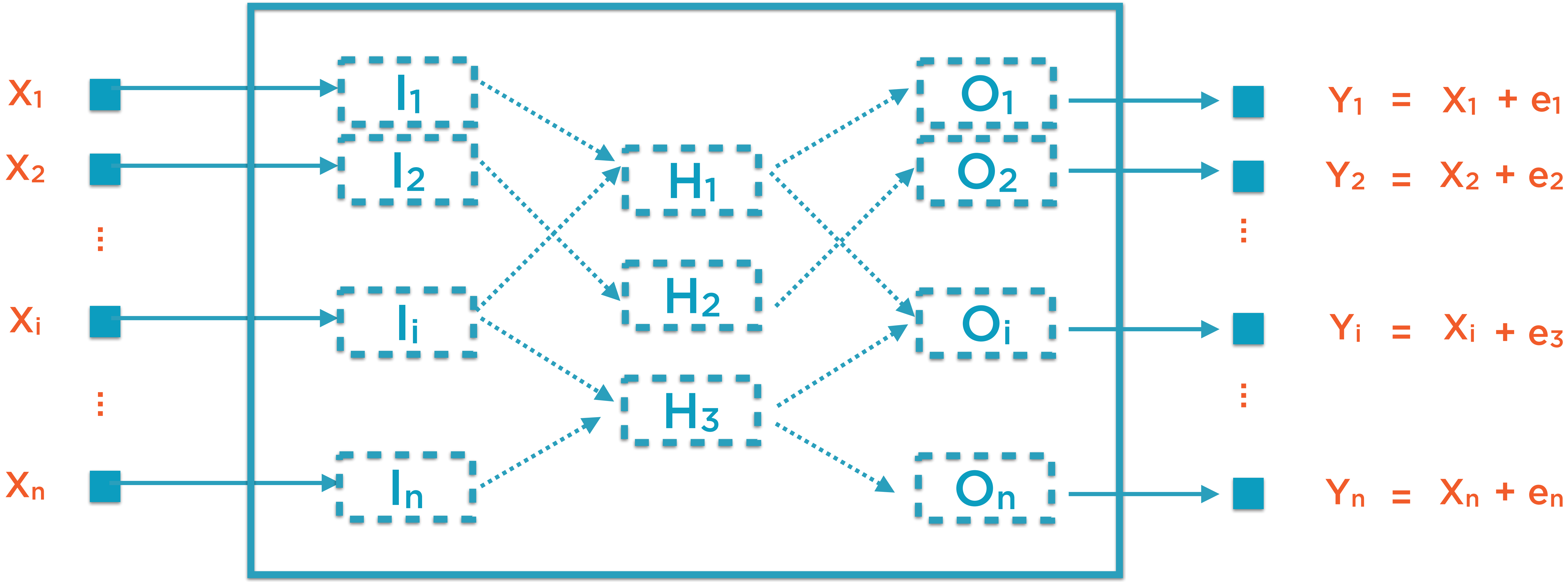
The autoencoder just passes the input through, so output is exactly equal to input

Trivial Autoencoder



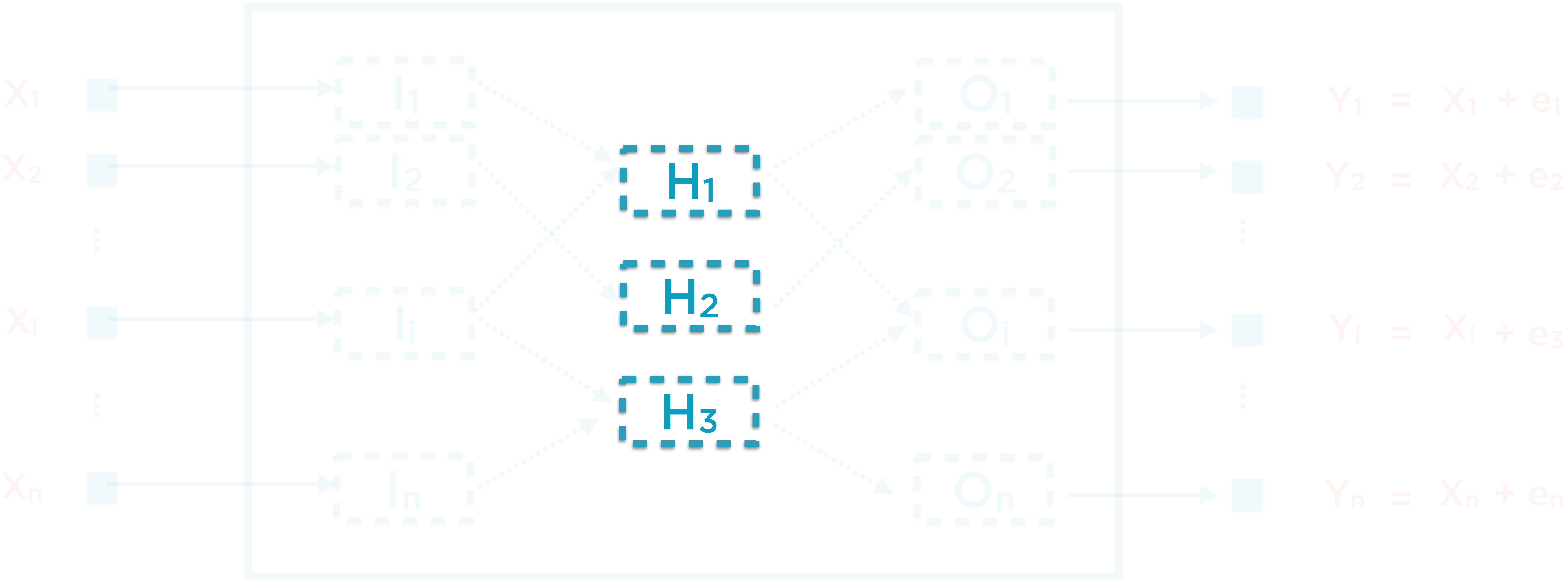
Now, let's constrain the network to force dimensionality reduction

Undercomplete Autoencoder



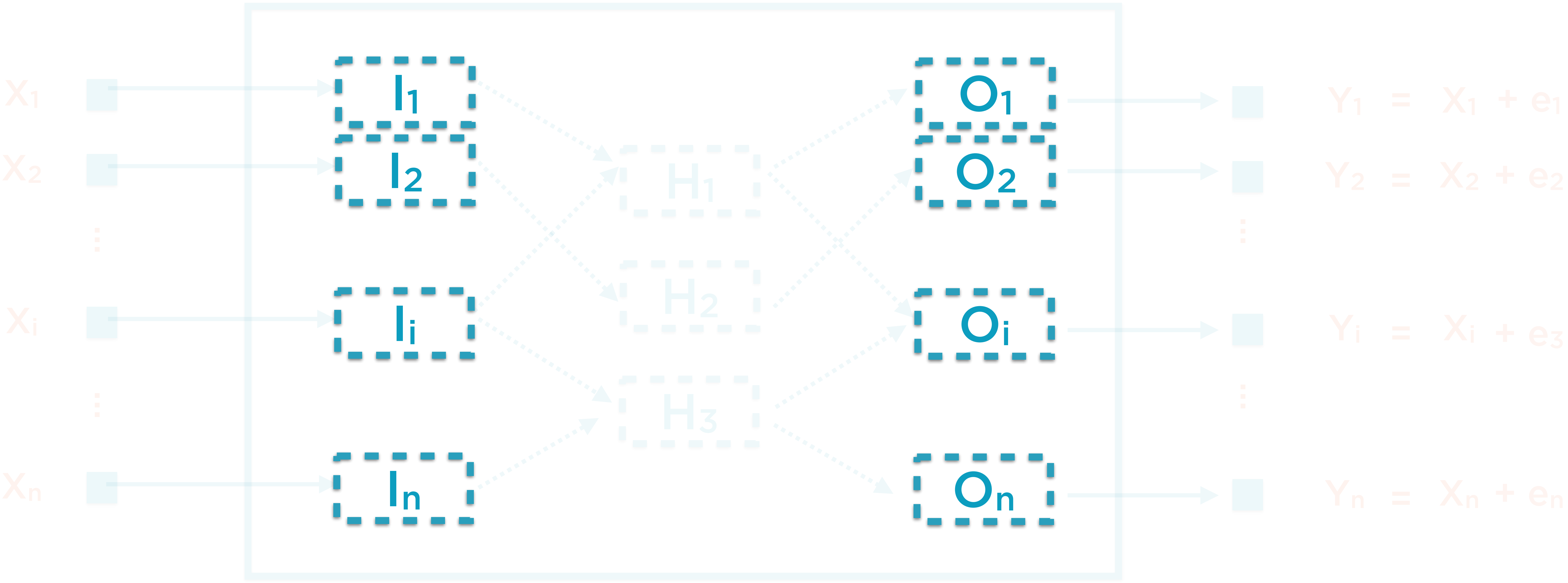
Dimensionality of the NN is now lower than that of input data

Undercomplete Autoencoder



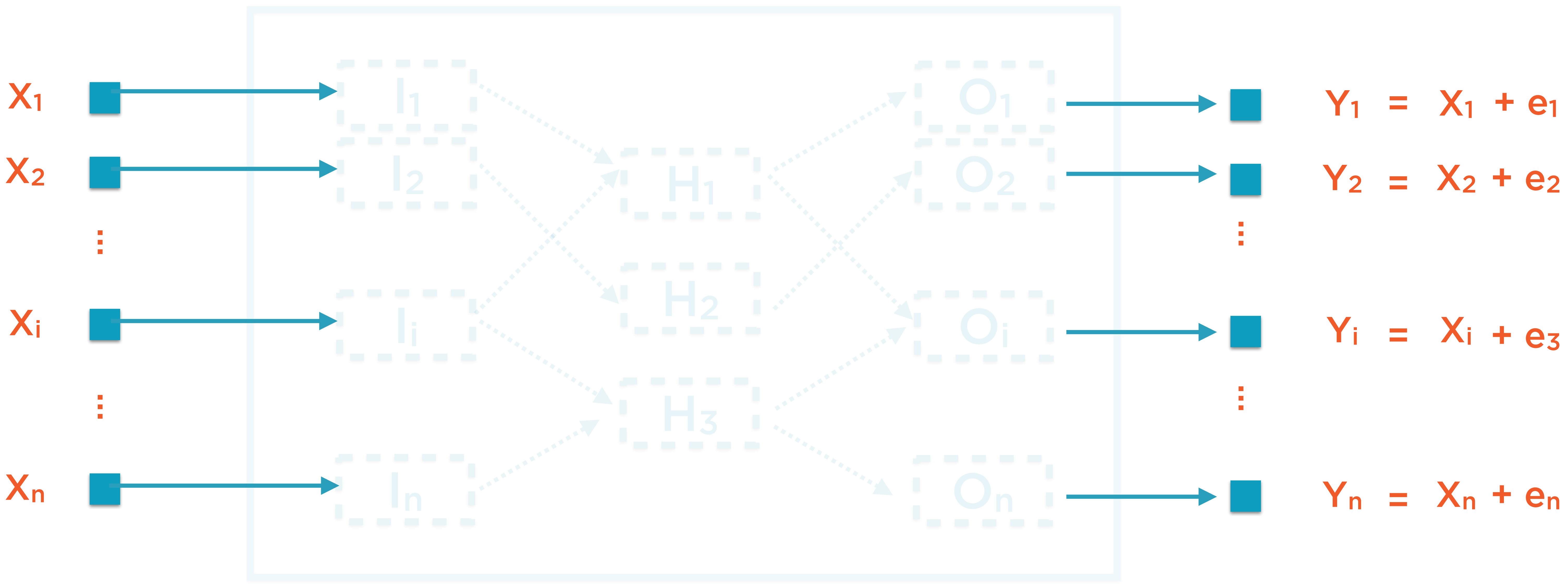
Add a middle, hidden layer with just three neurons (3 < N)

Undercomplete Autoencoder



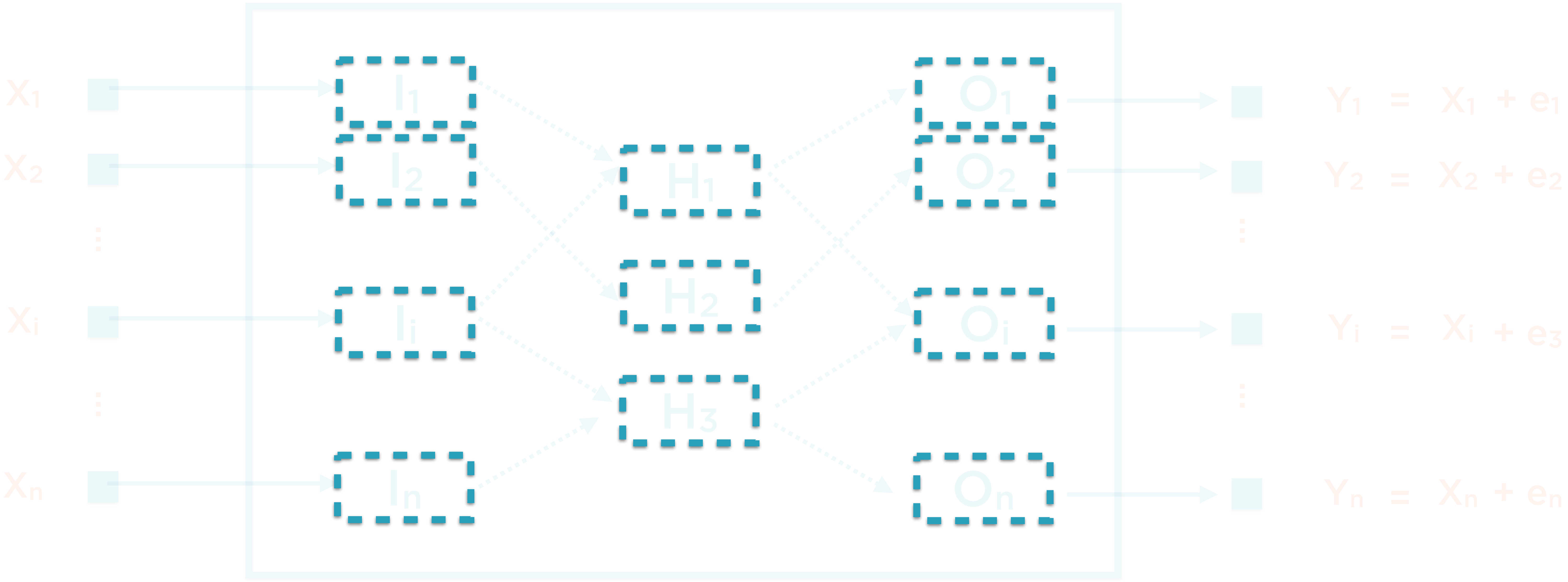
The input and output layers must now be separated, since each must still have same dimensionality as the input

Undercomplete Autoencoder



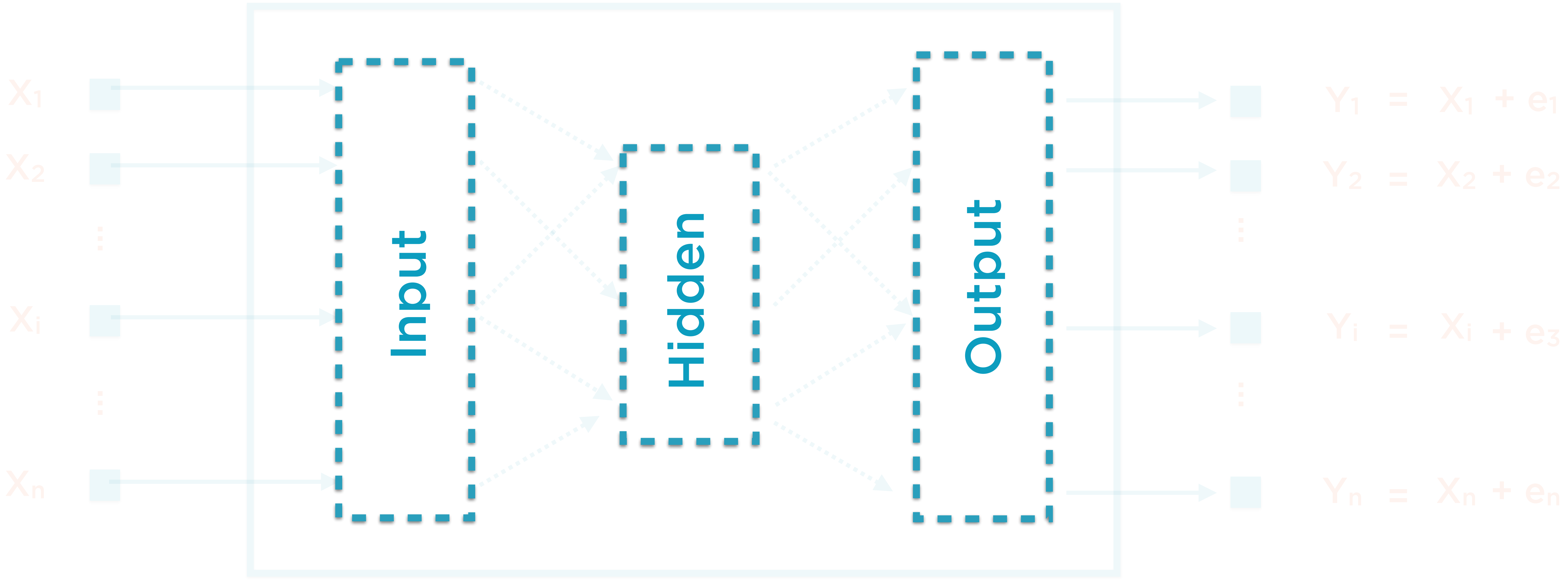
Why? Because autoencoder seeks to reconstruct input

Undercomplete Autoencoder



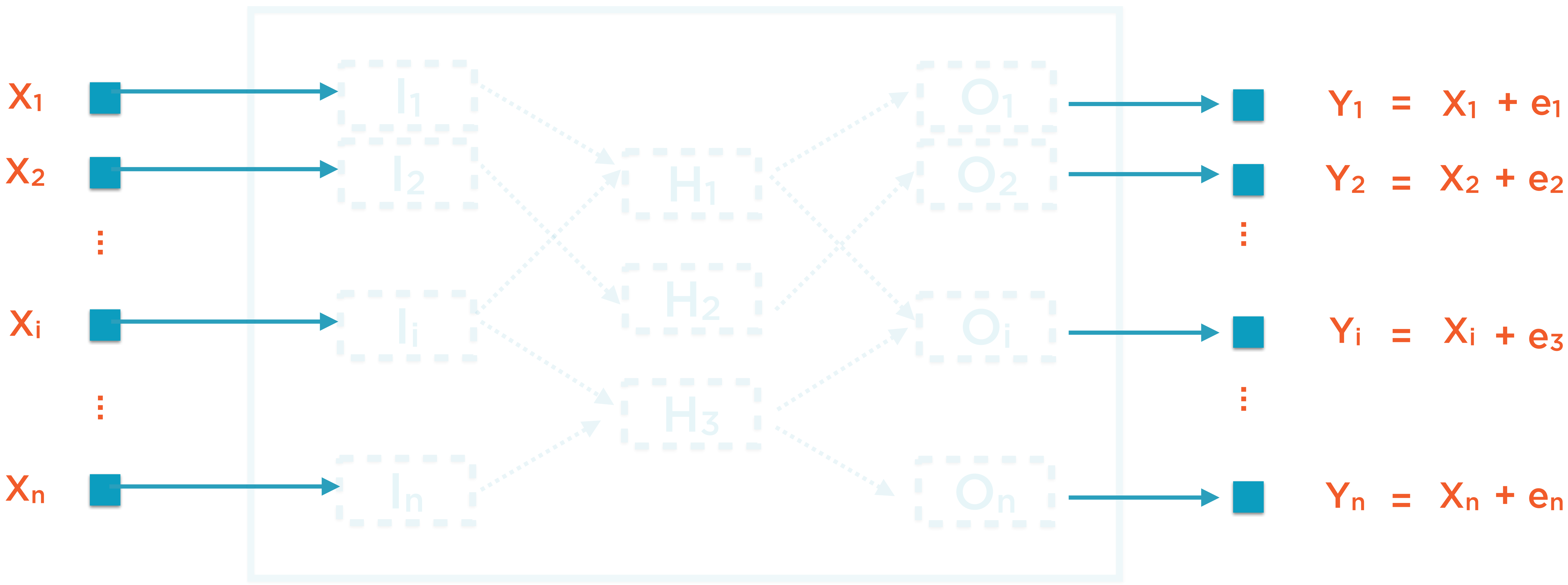
This gives undercomplete autoencoders a characteristic sandwich-like appearance

Undercomplete Autoencoder



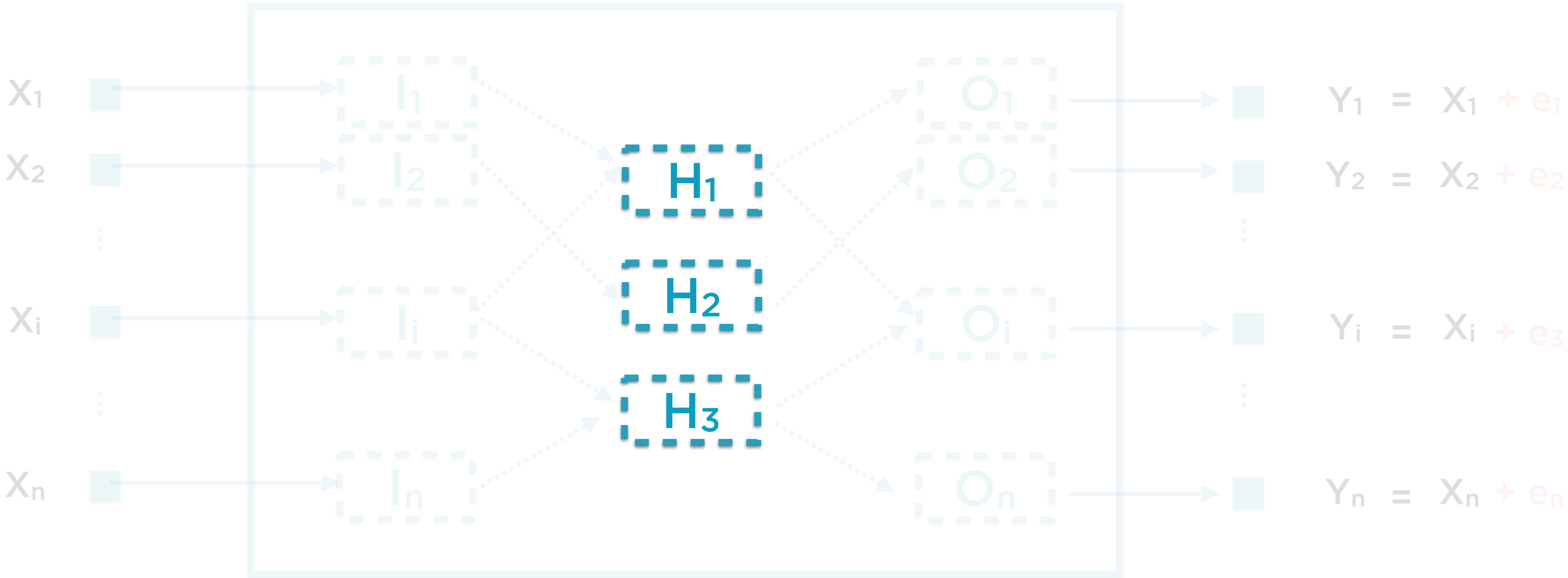
This gives undercomplete autoencoders a characteristic sandwich-like appearance

Undercomplete Autoencoder



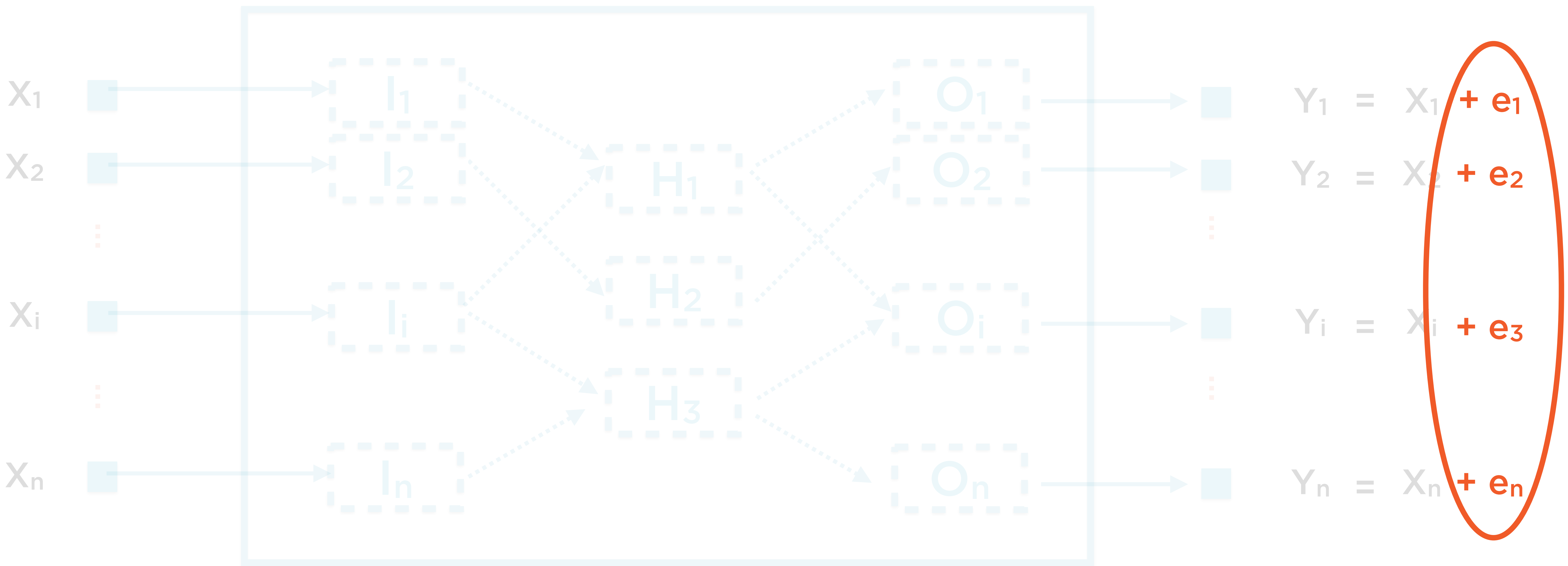
The undercomplete autoencoder will try to exactly match the input, but it will likely not succeed completely

Undercomplete Autoencoder



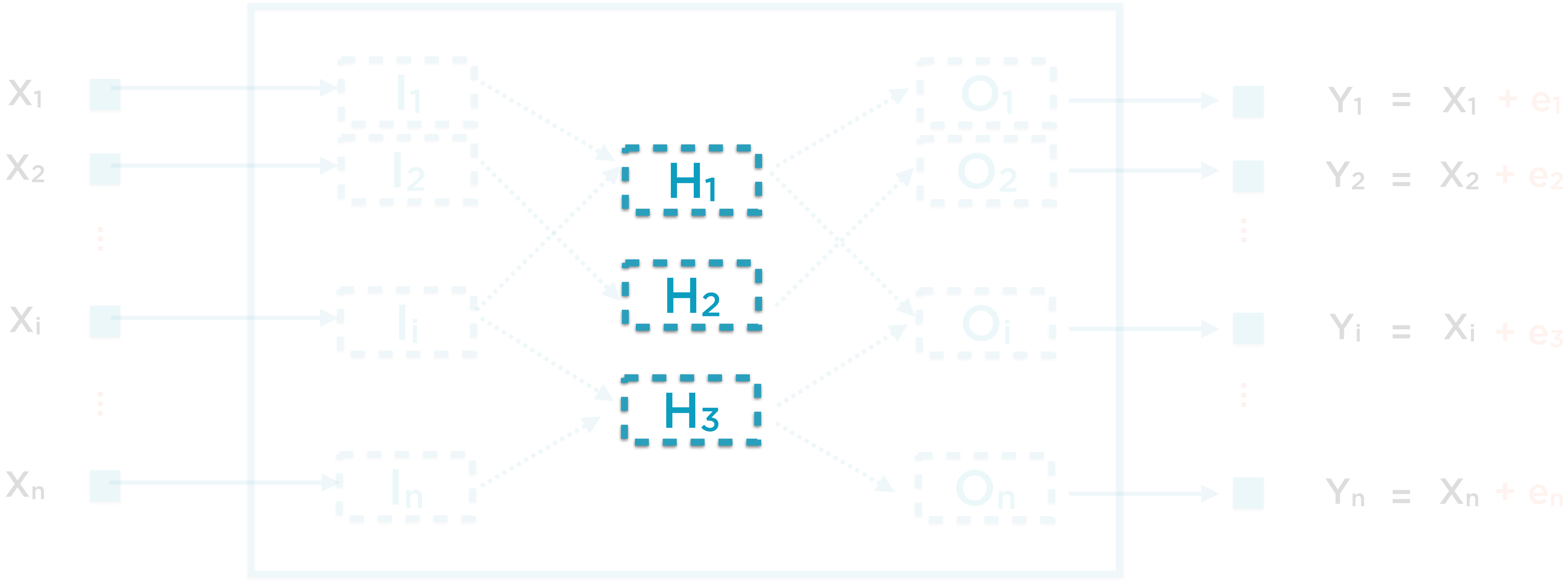
Now, because of the dimensionality reduction, output will not be exactly same as input

Undercomplete Autoencoder



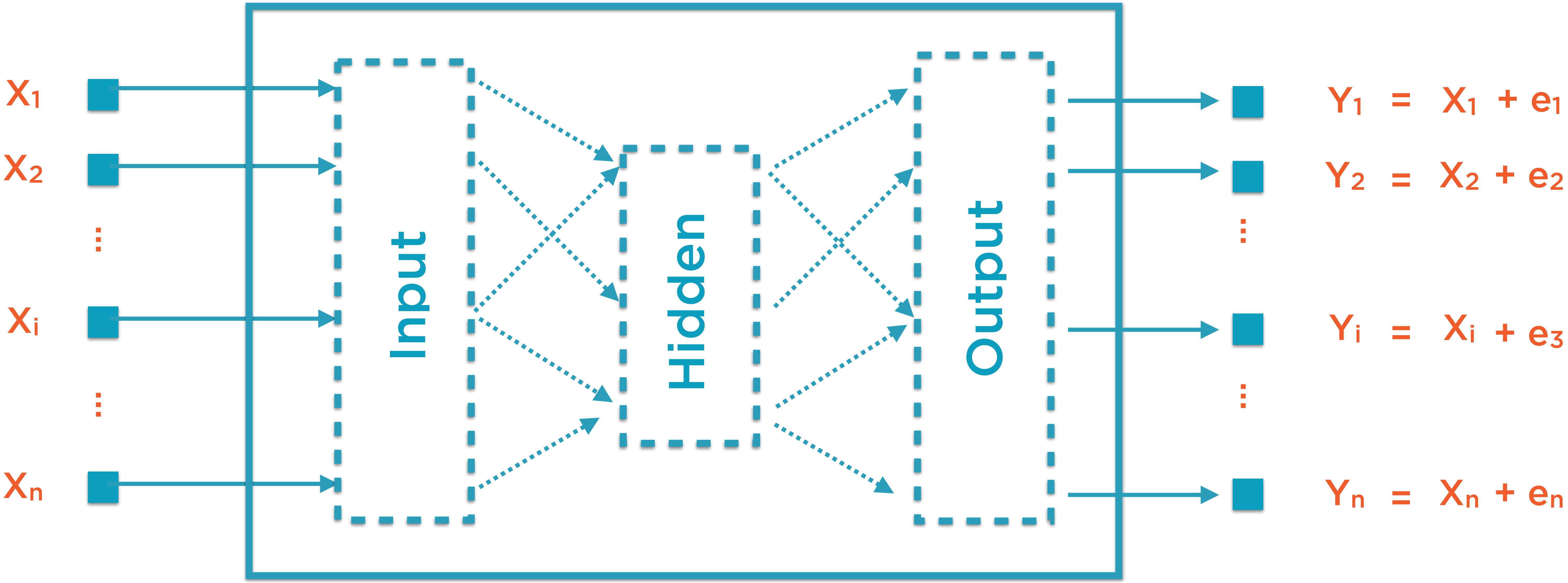
A reconstruction error will now exist

Undercomplete Autoencoder



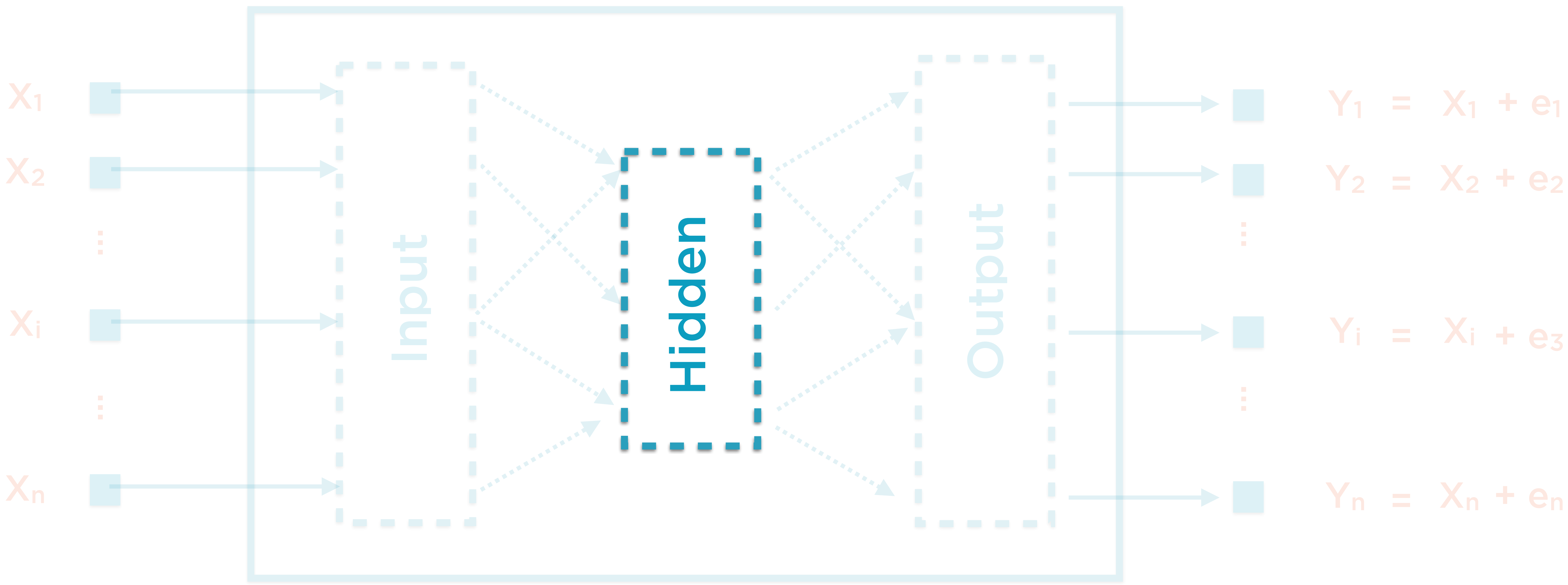
The autoencoder will be forced to learn the most significant characteristics of the data i.e. latent factors

Undercomplete Autoencoder



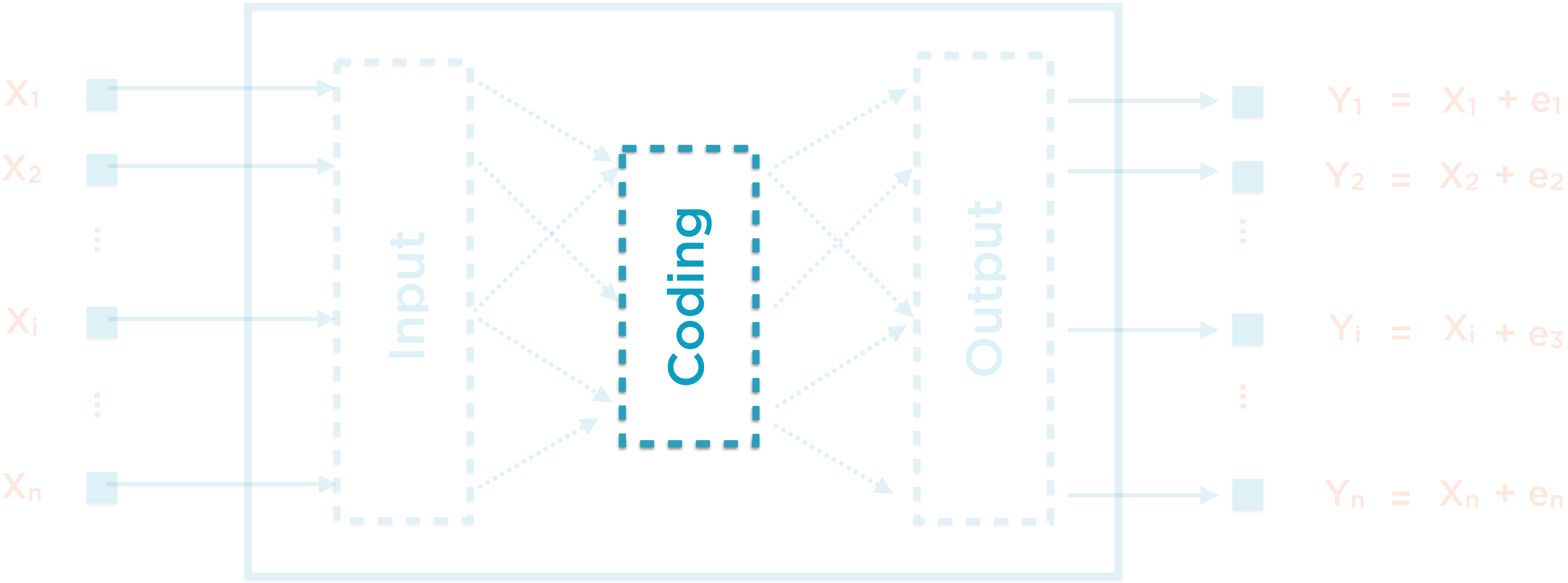
The central hidden layer is called the coding layer

Undercomplete Autoencoder



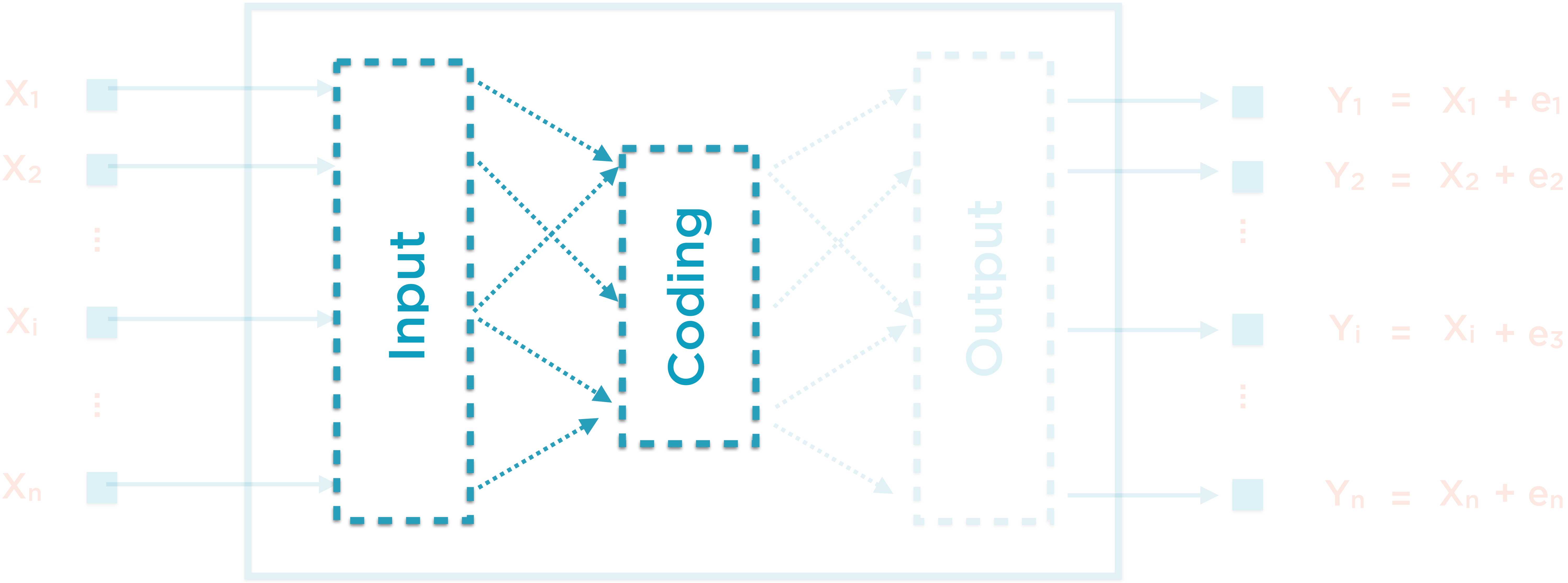
The central hidden layer is called the coding layer

Undercomplete Autoencoder

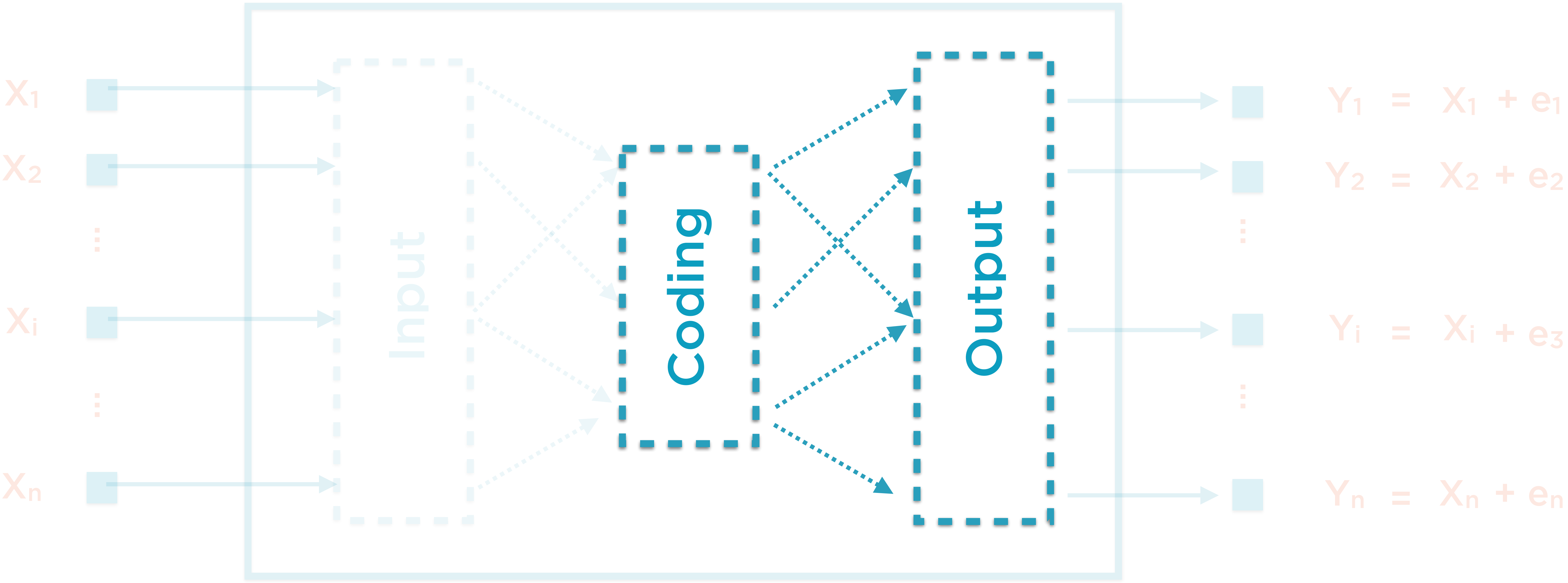


The central hidden layer is called the coding layer

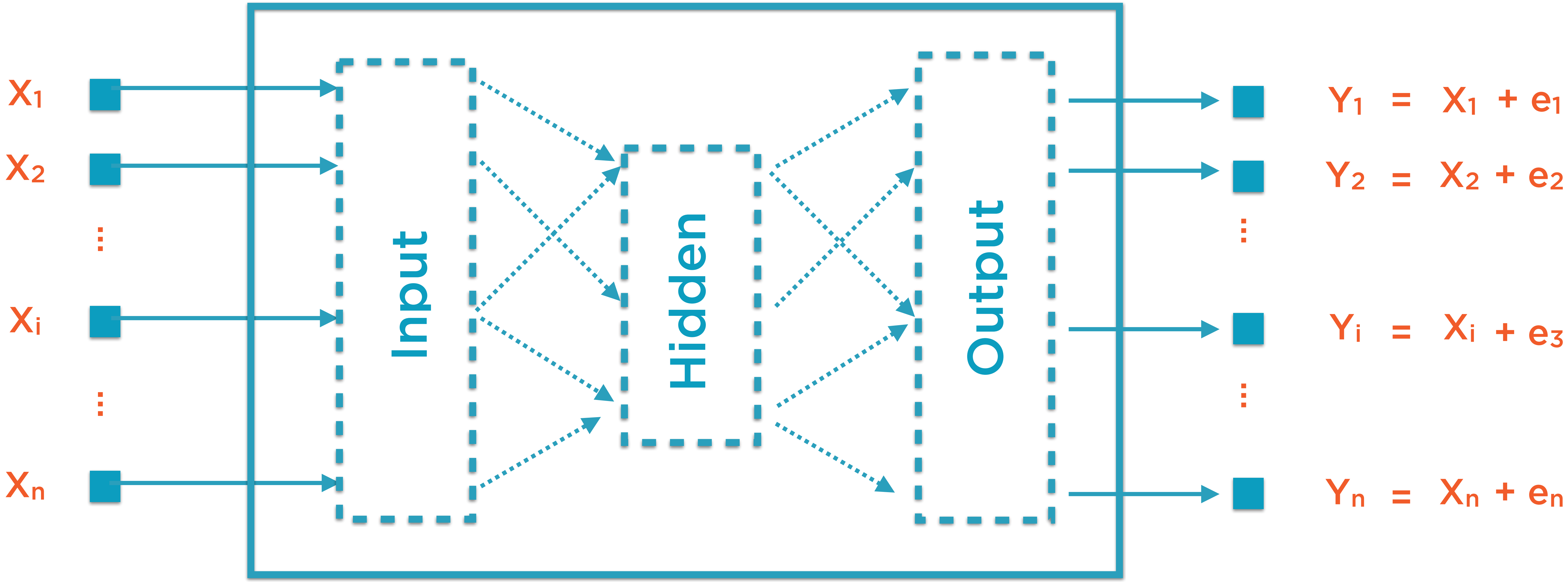
Encoding



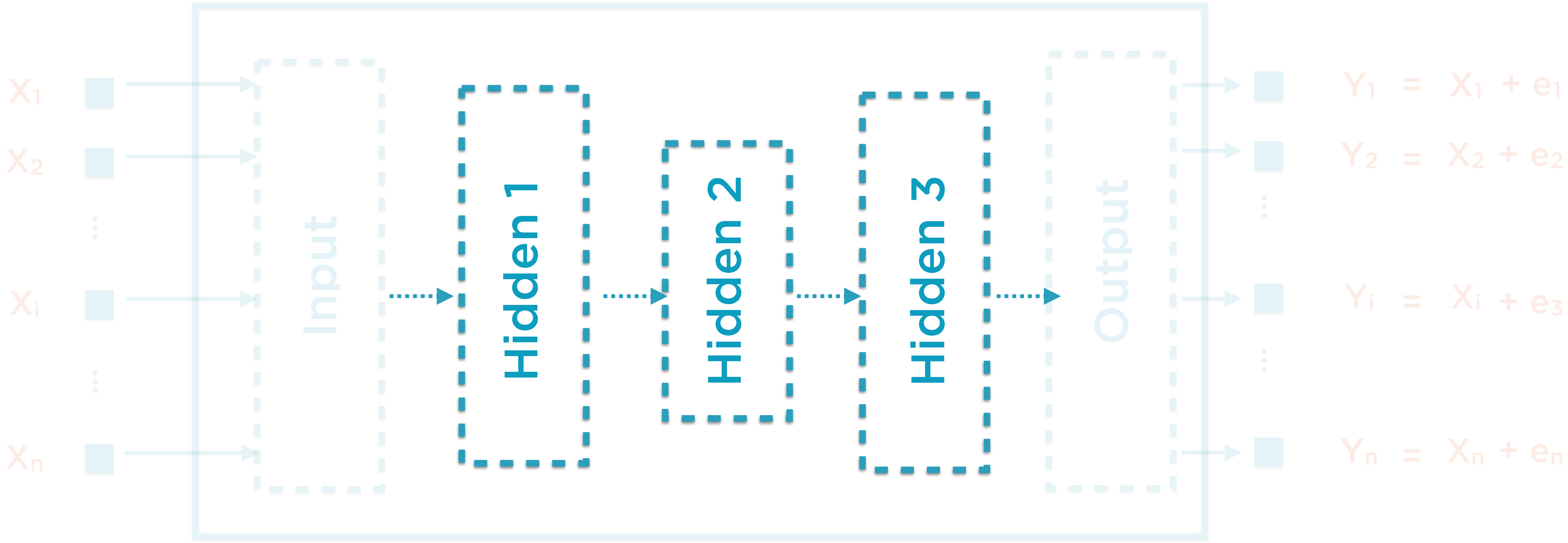
Decoding



Autoencoder

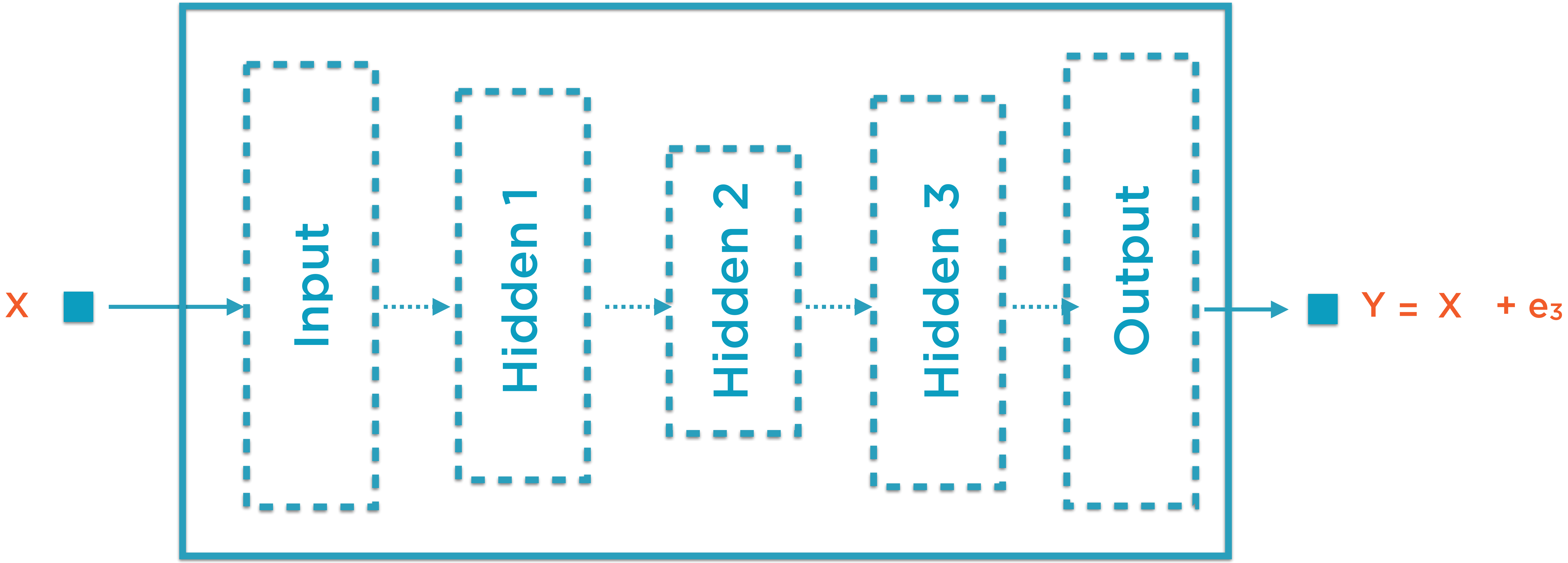


Stacked Autoencoder



Add multiple hidden layers to learn more complex internal patterns

Stacked Autoencoder



Add multiple hidden layers to learn more complex internal patterns

Demo

Unsupervised learning using autoencoders in Keras

Summary

Unsupervised machine learning models

Autoencoders to find latent features in data

Build and train an autoencoder using Keras layers

Up Next:

Implementing Custom Layers and Models