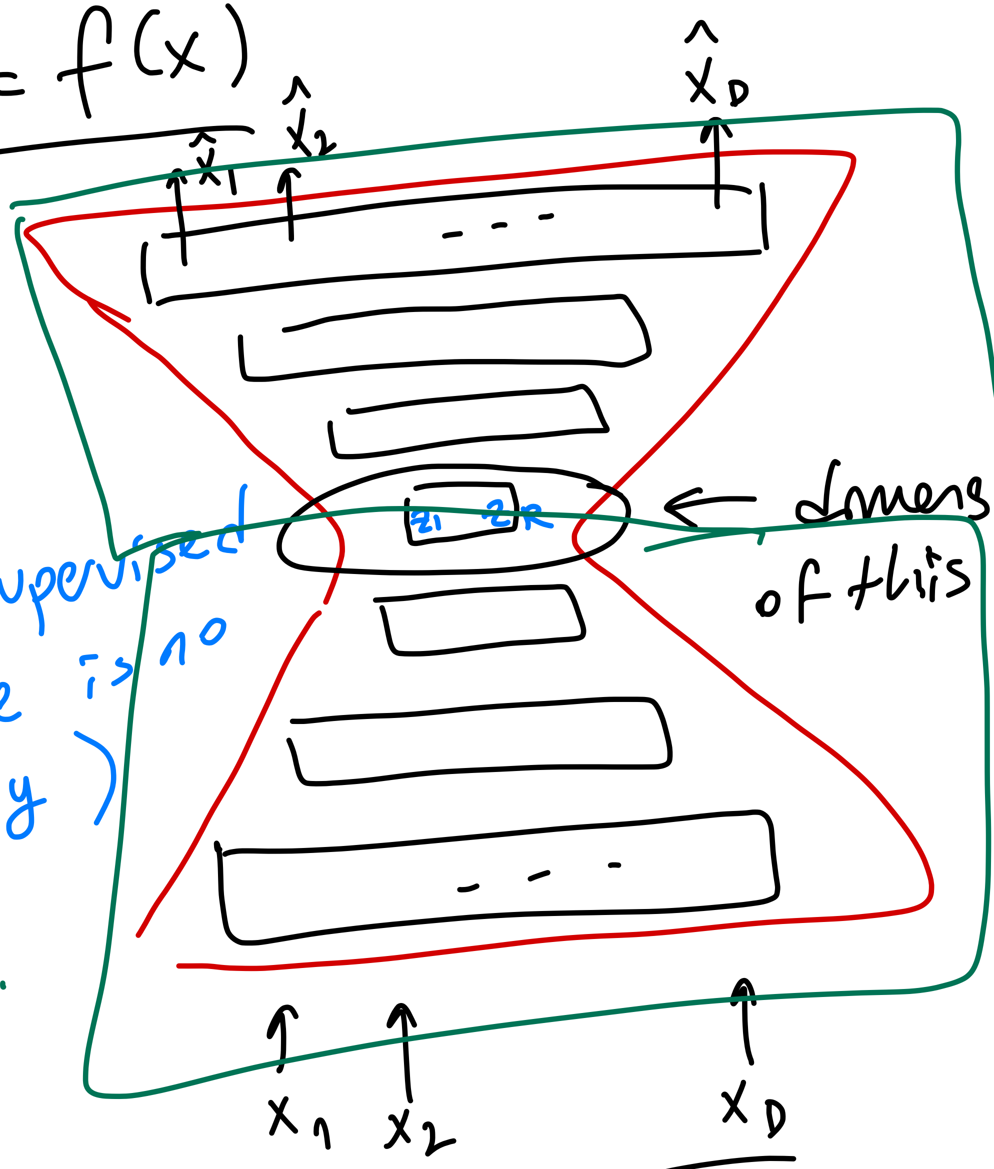
Autoencoders

$$\hat{y}_t = f(\underline{y_{t-1}}, \underline{y_{t-2}}, \dots, \underline{y_{t-10}})$$

$\underbrace{\qquad\qquad\qquad\qquad}_{\text{autoregressive}}$

$\hat{x} = f(x)$



decoder

encoder.

unsupervised (there is no y)

← dimensionality of this layer (R)

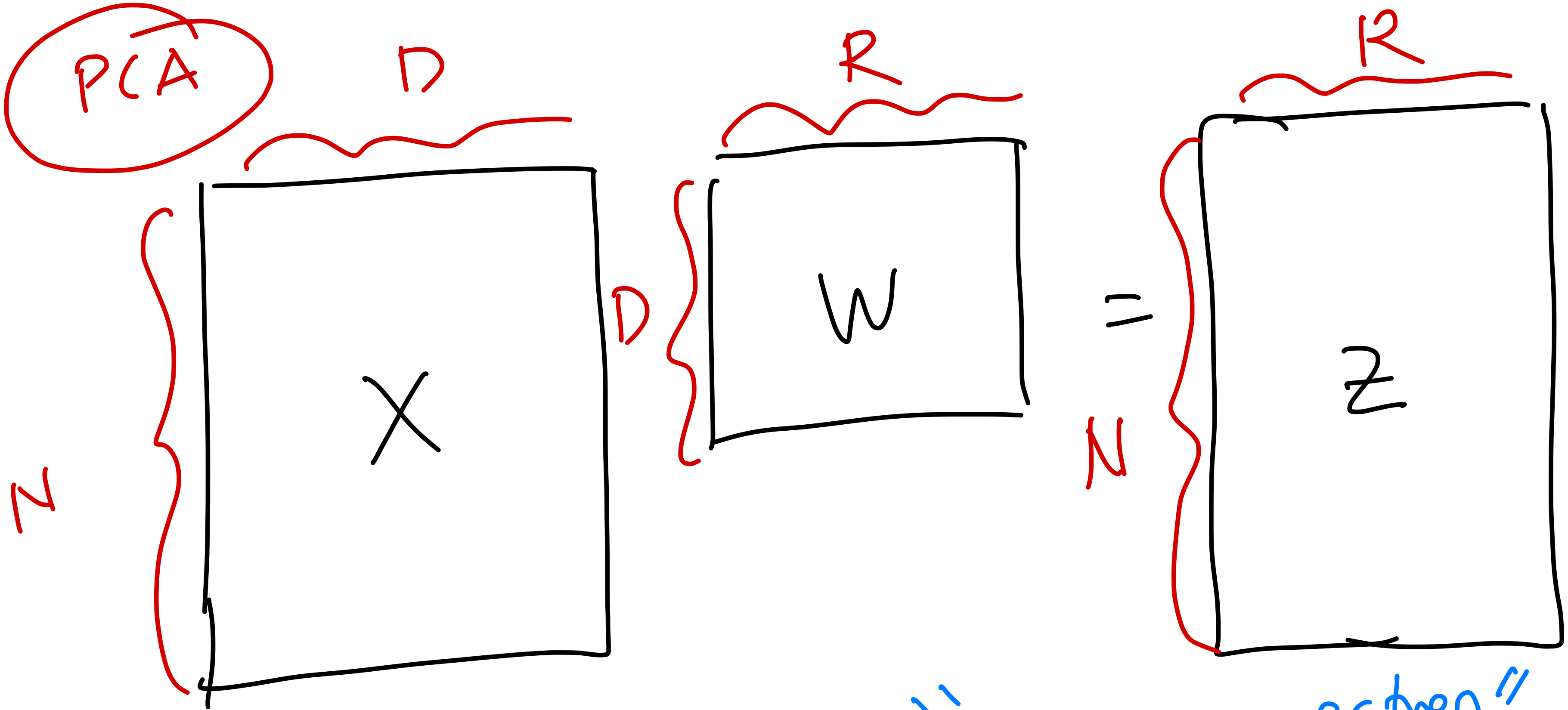
$x \in \mathbb{R}^D \Rightarrow$ input vector

They are similar to dimensionality reduction algorithms.

$R \ll D$

$$\underbrace{x \in \mathbb{R}^D \rightarrow z}_{\text{encoder}} \in \underbrace{\mathbb{R}^R \rightarrow \hat{x}}_{\text{decoder}} \in \mathbb{R}^D$$

PCA

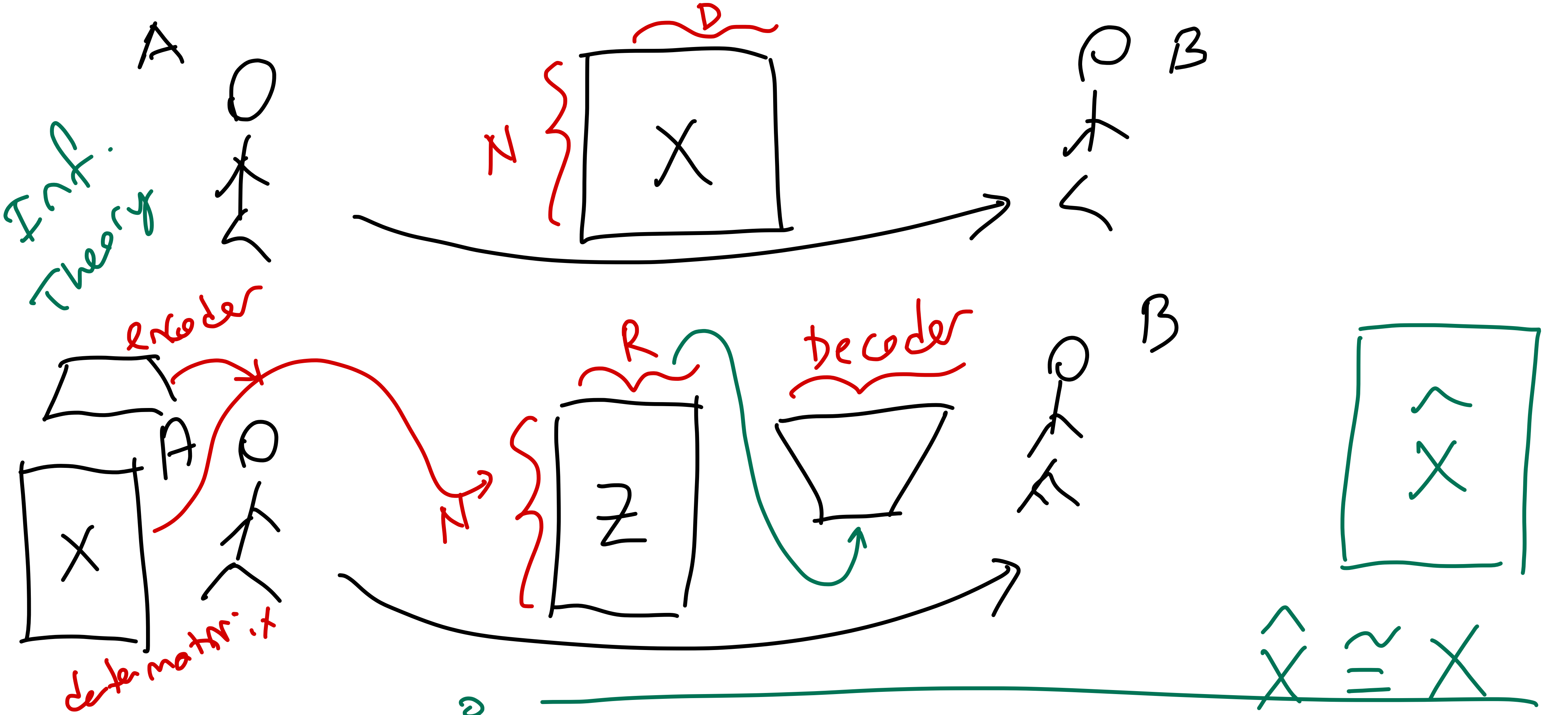


"Linear projection"

Loss Function (Reconstruction ~~Error~~)

the difference between x & $\hat{x}$.

$$\hat{X} \simeq X \Rightarrow \overset{m}{\cancel{\$}}se \Rightarrow \frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\left(X_{id}-\hat{X}_{id}\right)^2$$



$\hat{X} \cong X$

visualization

if $R=2$ or $R=3$

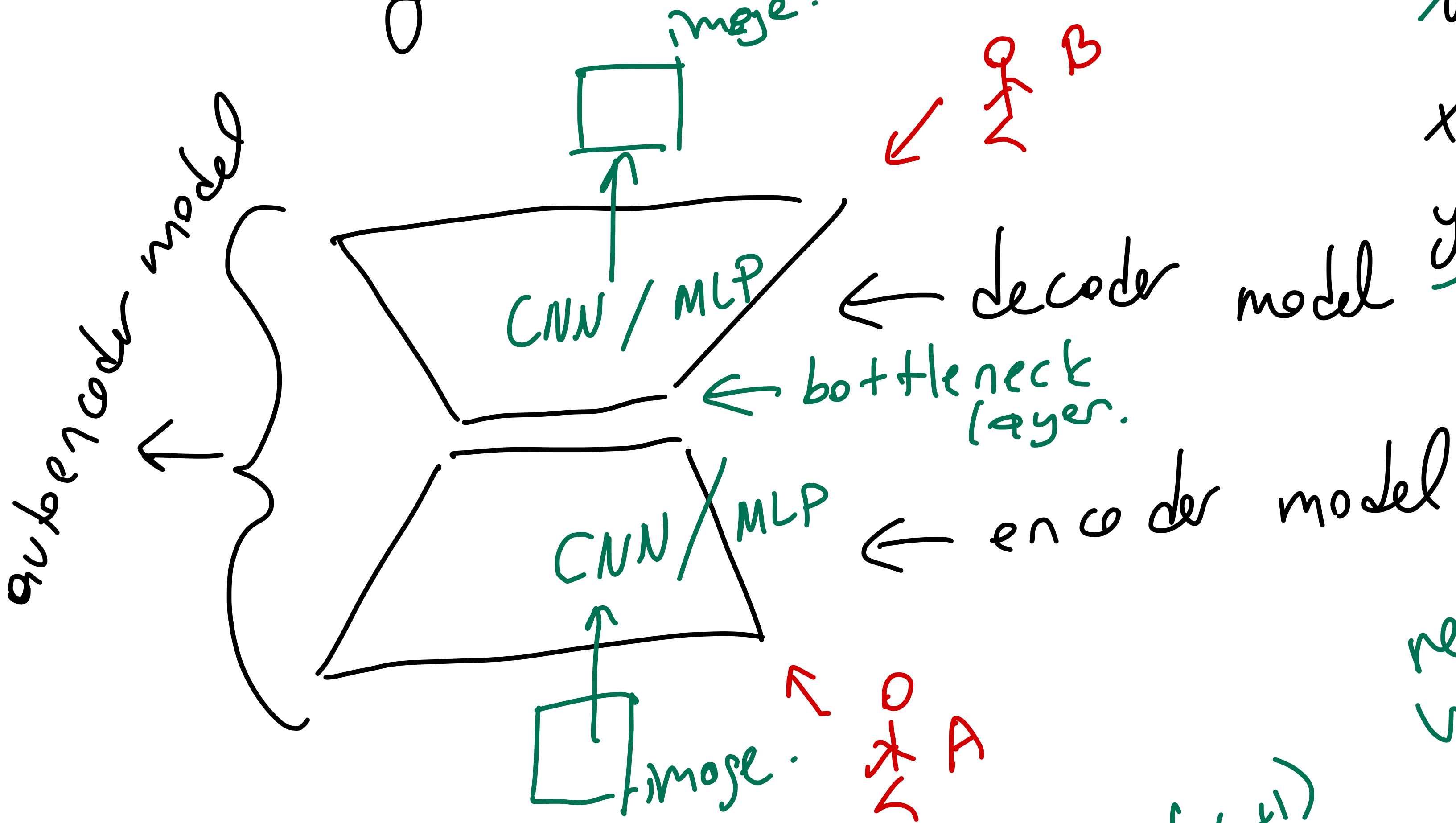
$\Downarrow$

We can use Z to visualize data.

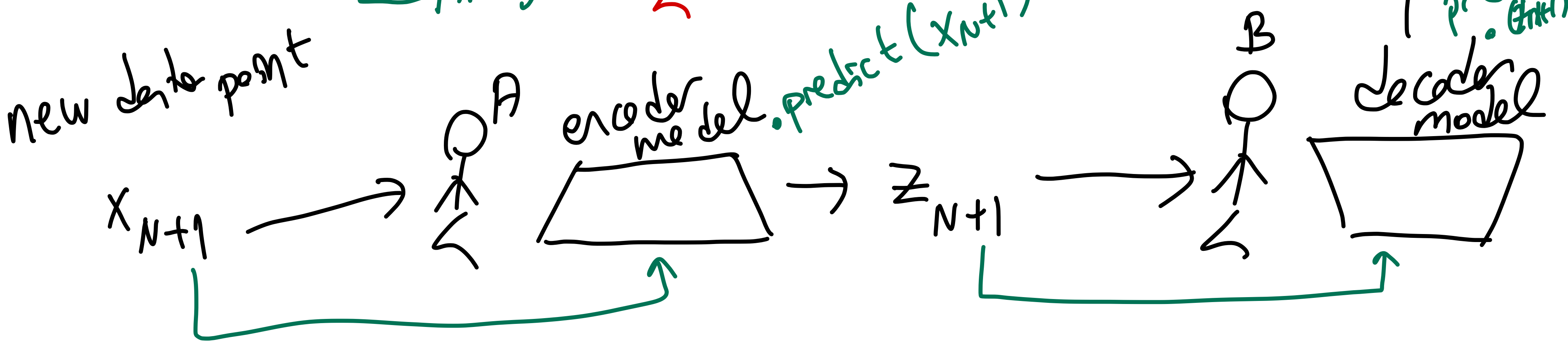
Multioutput regression where $y_i = x_i$ → $\{(x_i, \cancel{y_i})\}_{i=1}^{N}$ (x_i)

$x_i \in \mathbb{R}^D$

$\cancel{y_i \in \mathbb{R}^Q}$ $x_i \in \mathbb{R}^D$



reconstructed version of x_{N+1} ⇒ $\hat{x}_{N+1}$

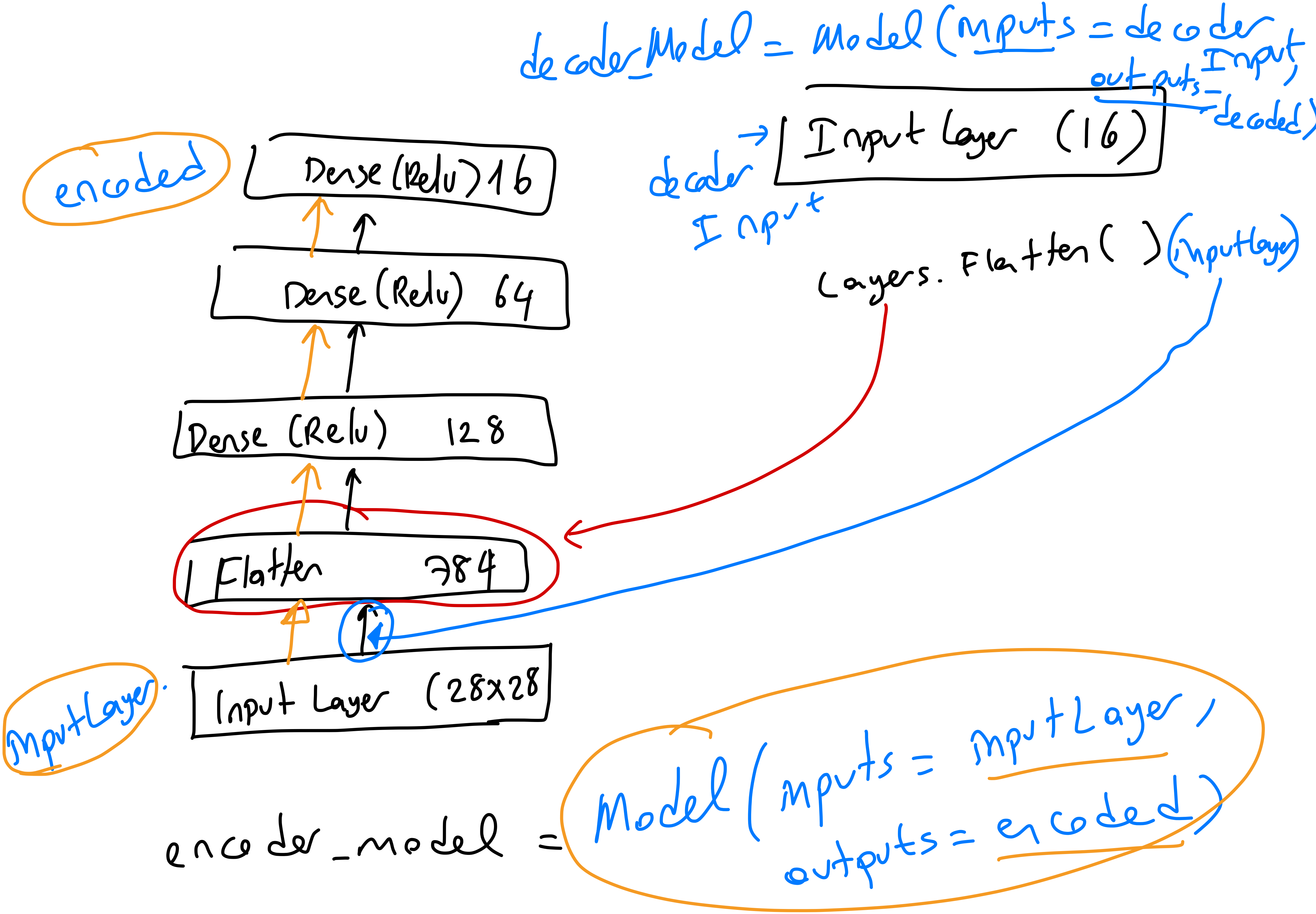


build encoder model
build decoder model
combine them into autoencoder model
Compile & fit autoencoder model.
call predict of encoder model (A)
call predict of decoder model (B)

$\underline{x_i} \Rightarrow z_i \Longrightarrow (z_i + \text{add noise}) \Rightarrow$ decoder

$\Downarrow$

perturbed version of x_i.



decoder_Model = Model(inputs = decoder Input, outputs = decoded)
Input Layer (16)
decoder Input
Layers. Flatten( )(inputLayer)
encoded
Dense (Relu) 16
Dense (Relu) 64
Dense (Relu) 128
Flatten 784
inputLayer.
Input Layer (28x28)
encoder_model = Model(inputs = inputLayer, outputs = encoded)