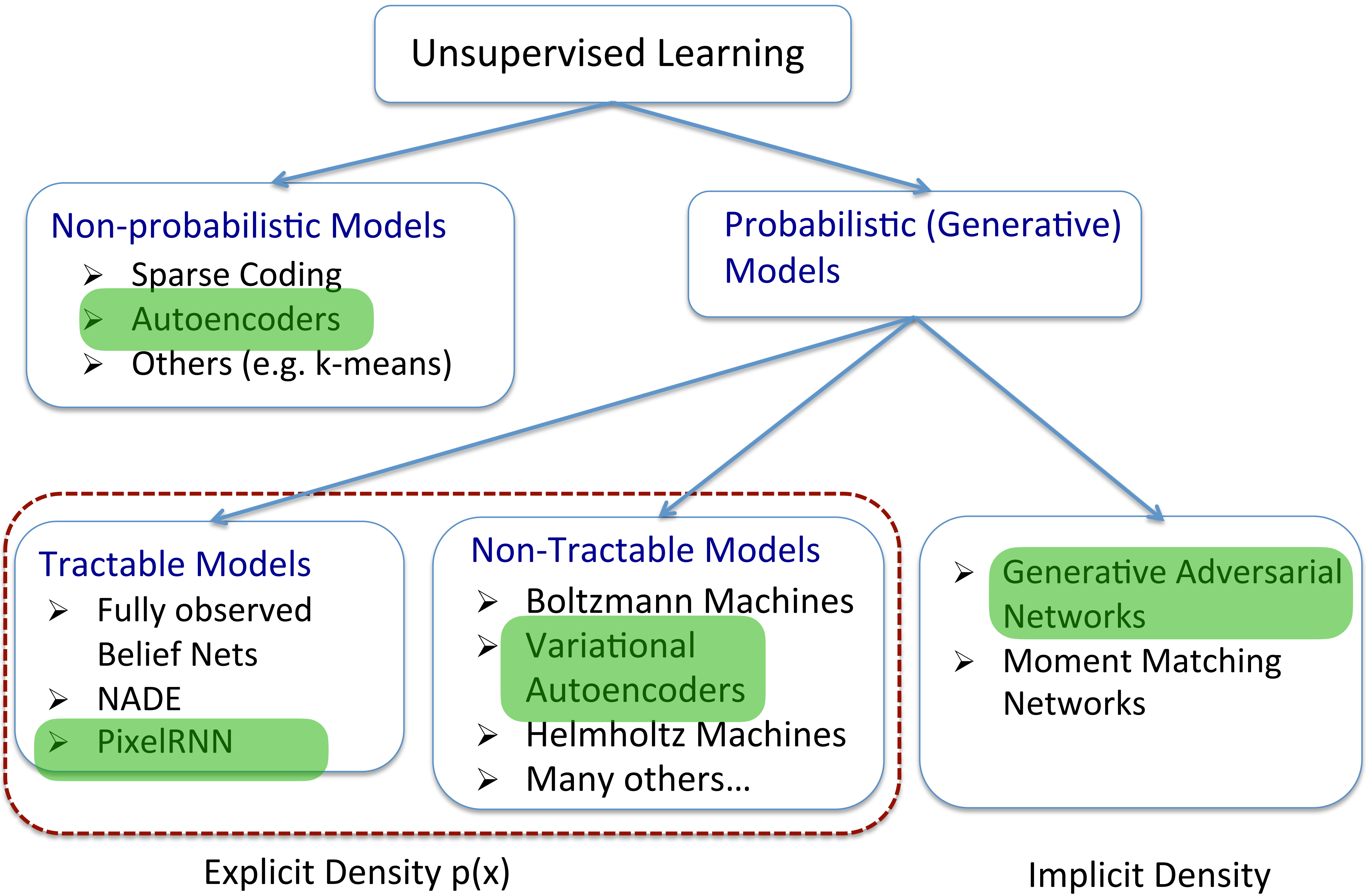
Deep Learning with Neural Networks

Unsupervised Deep Learning (I): autoencoders

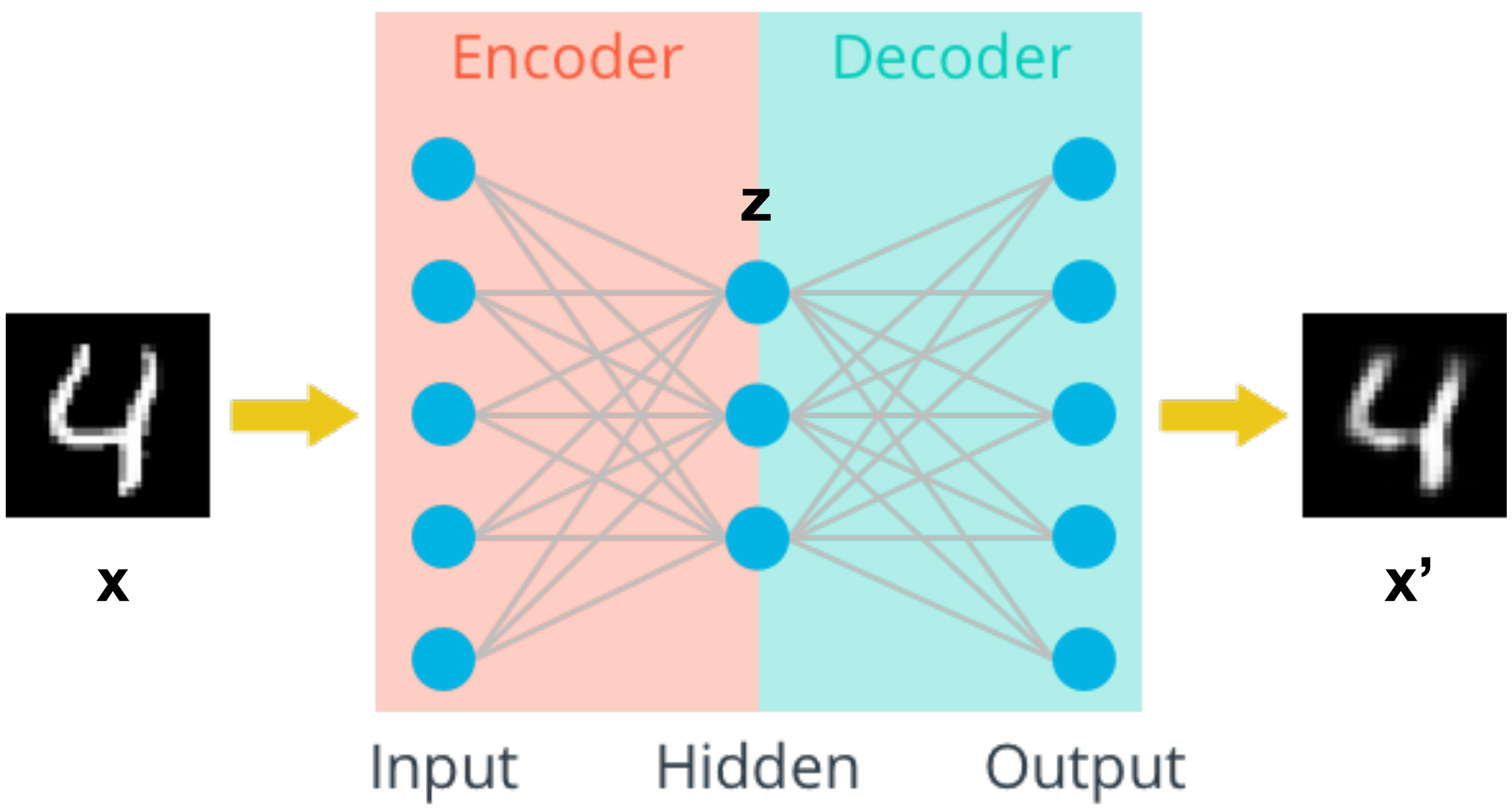
Original: Pablo Martínez Olmos, pamartin@ing.uc3m.es
Adapted: Alejandro Guerrero López, alguerre@pa.uc3m.es



Unsupervised Learning
Non-probabilistic Models
Sparse Coding
Autoencoders
Others (e.g. k-means)
Probabilistic (Generative) Models
Tractable Models
Fully observed Belief Nets
NADE
PixelRNN
Non-Tractable Models
Boltzmann Machines
Variational Autoencoders
Helmholtz Machines
Many others…
Generative Adversarial Networks
Moment Matching Networks
Explicit Density p(x)
Implicit Density

Deep Autoencoder

- An autoencoder is a neural network that is trained to attempt to copy its input to its output
- Internally, it has a low-dimensional hidden representation
- Two parts: encoder **f(x)** and decoder **g(z)**:
  - where **x** is the input data, e.g., an image of a four
  - where **z** = **f(x)**, i.e., the output of the encoder
- Autoencoders have been traditionally used for dimensionality reduction or feature learning



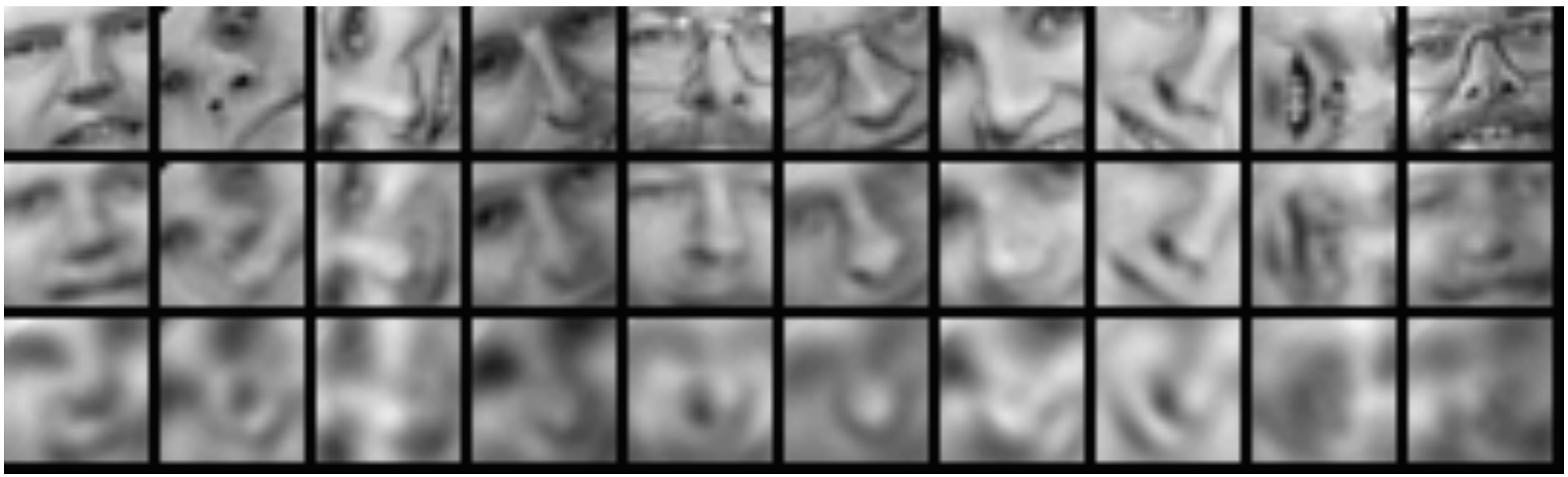
Encoder

$$L(\boldsymbol{x}, g(f(\boldsymbol{x})))$$

Decoder

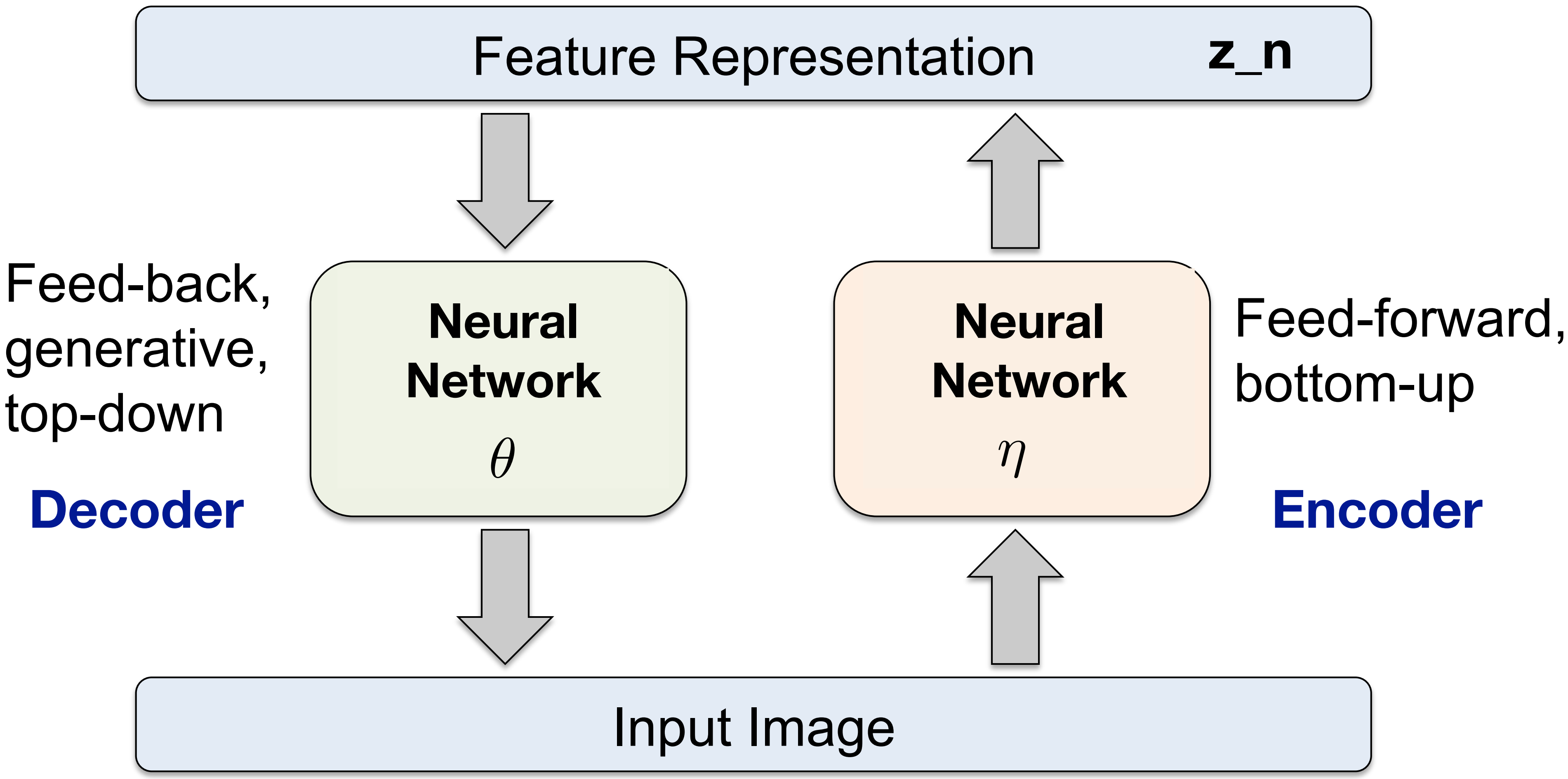
Deep Autoencoder

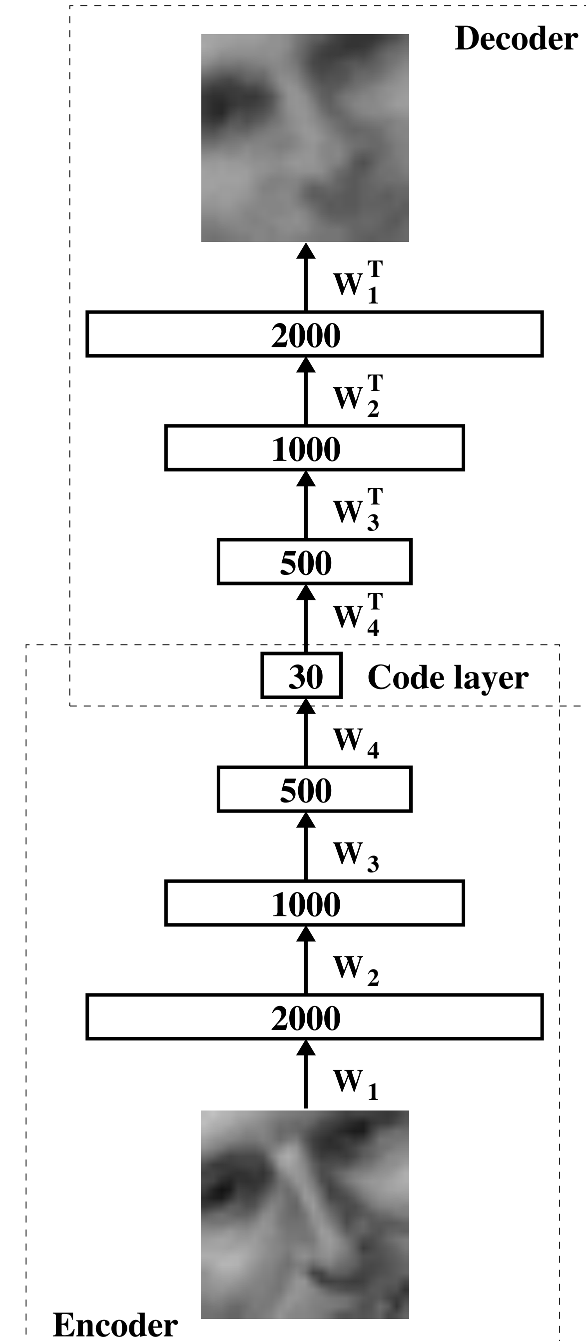
- 25x25 – 2000 – 1000 – 500 – 30 autoencoder to extract 30-D real-valued codes for Olivetti face patches.



- **Top**: Random samples from the test dataset.
- **Middle**: Reconstructions by the 30-dimensional deep autoencoder.
- **Bottom**: Reconstructions by the 30-dimentinoal PCA.

Deep Autoencoders



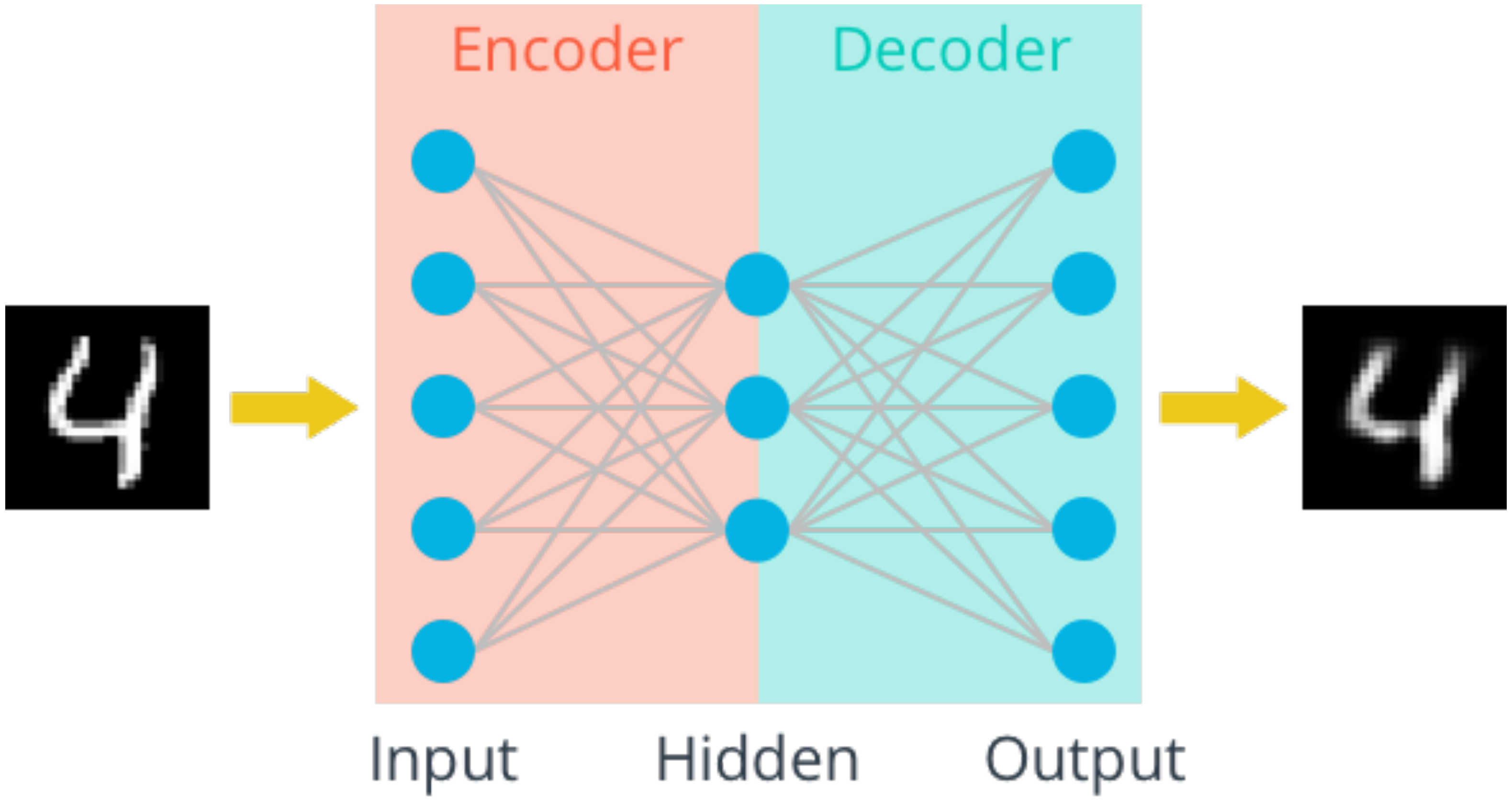


Encoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta (\mathbf{x}_n) \right) \right)$$

Decoder

Sparse Deep Autoencoders



Encoder

$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h})$$

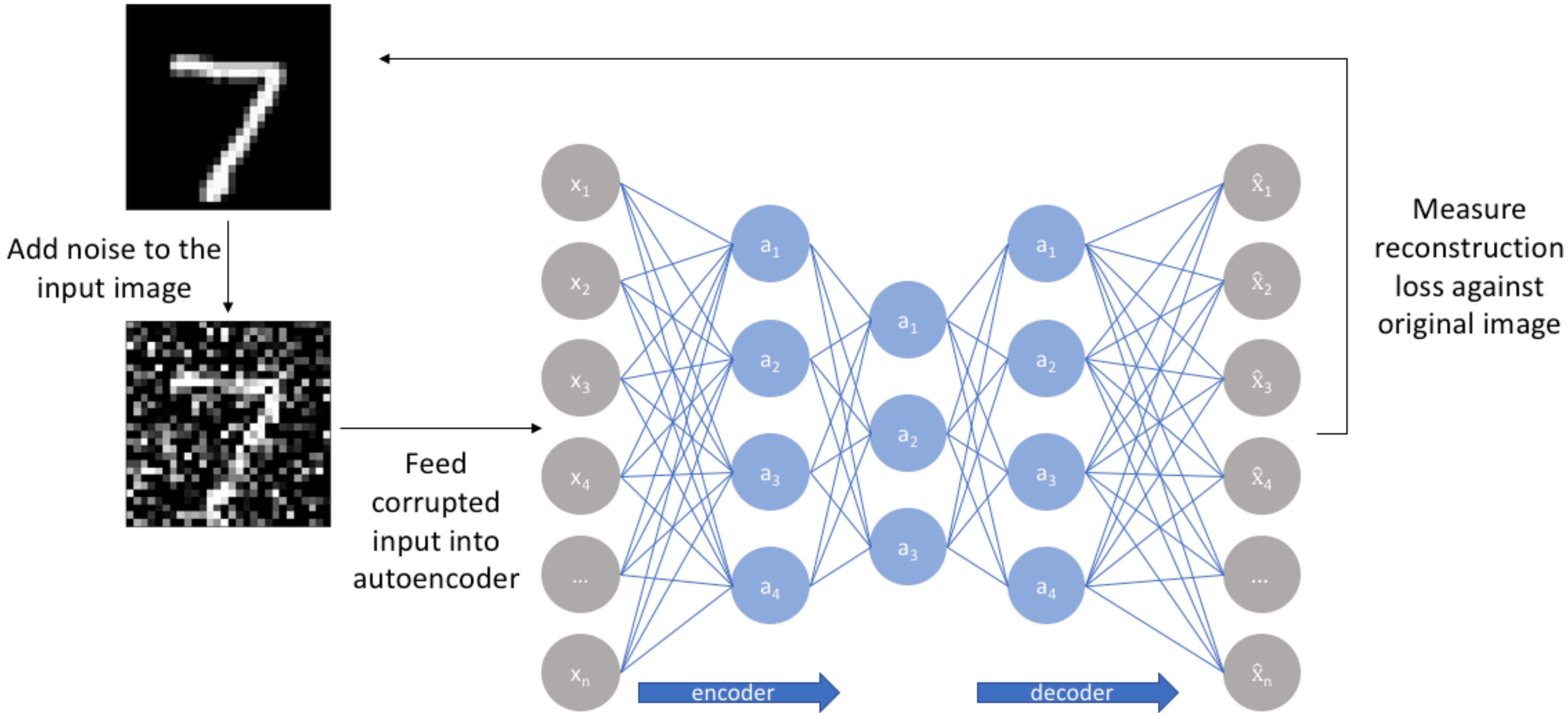
Decoder

Sparsity penalizer

$$\Omega(\boldsymbol{h}) = \lambda \sum_i |h_i|$$

- Sparse autoencoders are typically used to learn features for another task

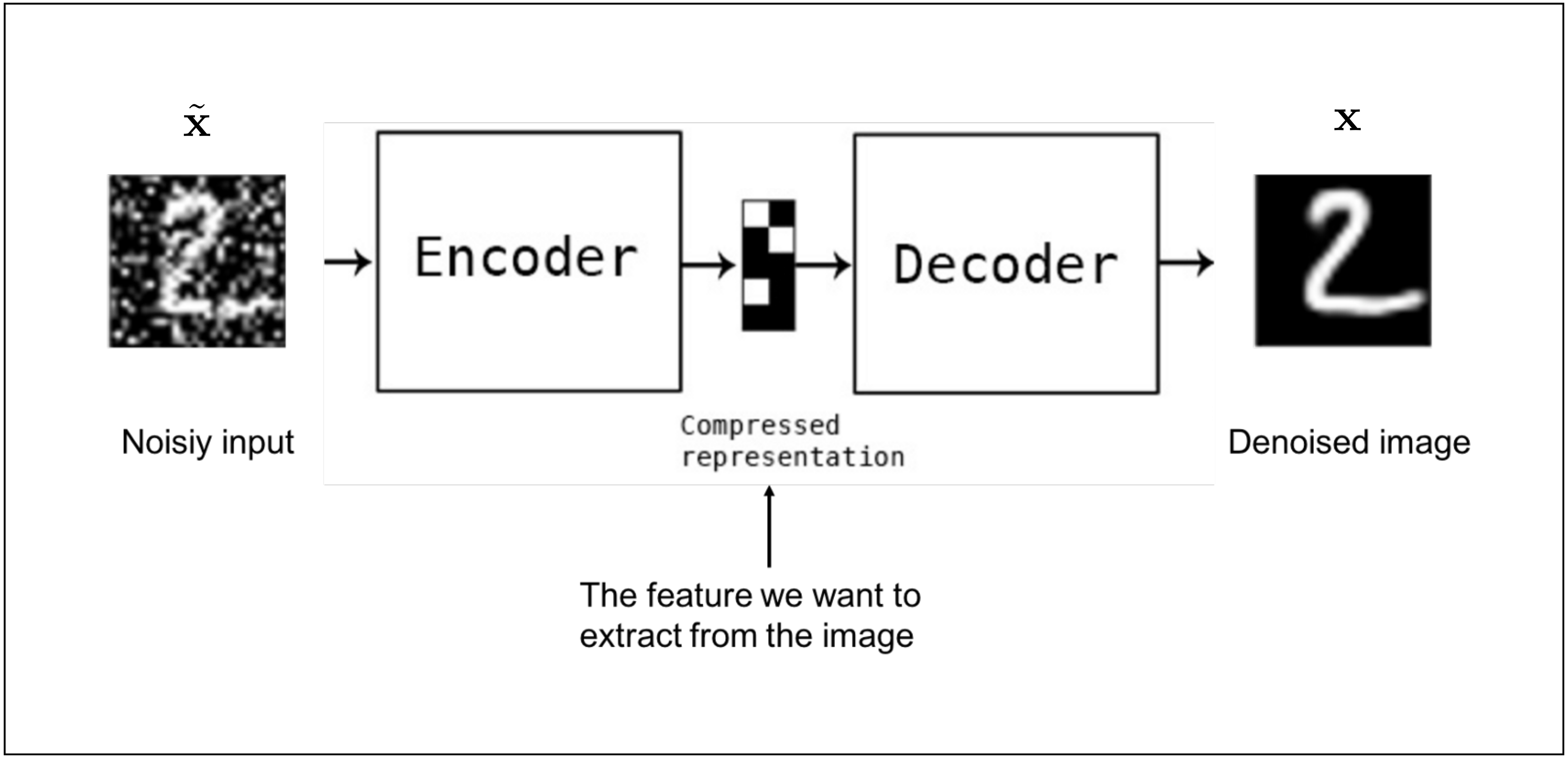
Denoising Deep Autoencoders



Source: this excellent blog

- Reconstruct a **corrupted version** of an image
- More robust solutions. It is some sort of regularization
- They are widely used for image denoising and missing data completion

Denoising Deep Autoencoders



$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\tilde{\mathbf{x}}_n) \right) \right)$$