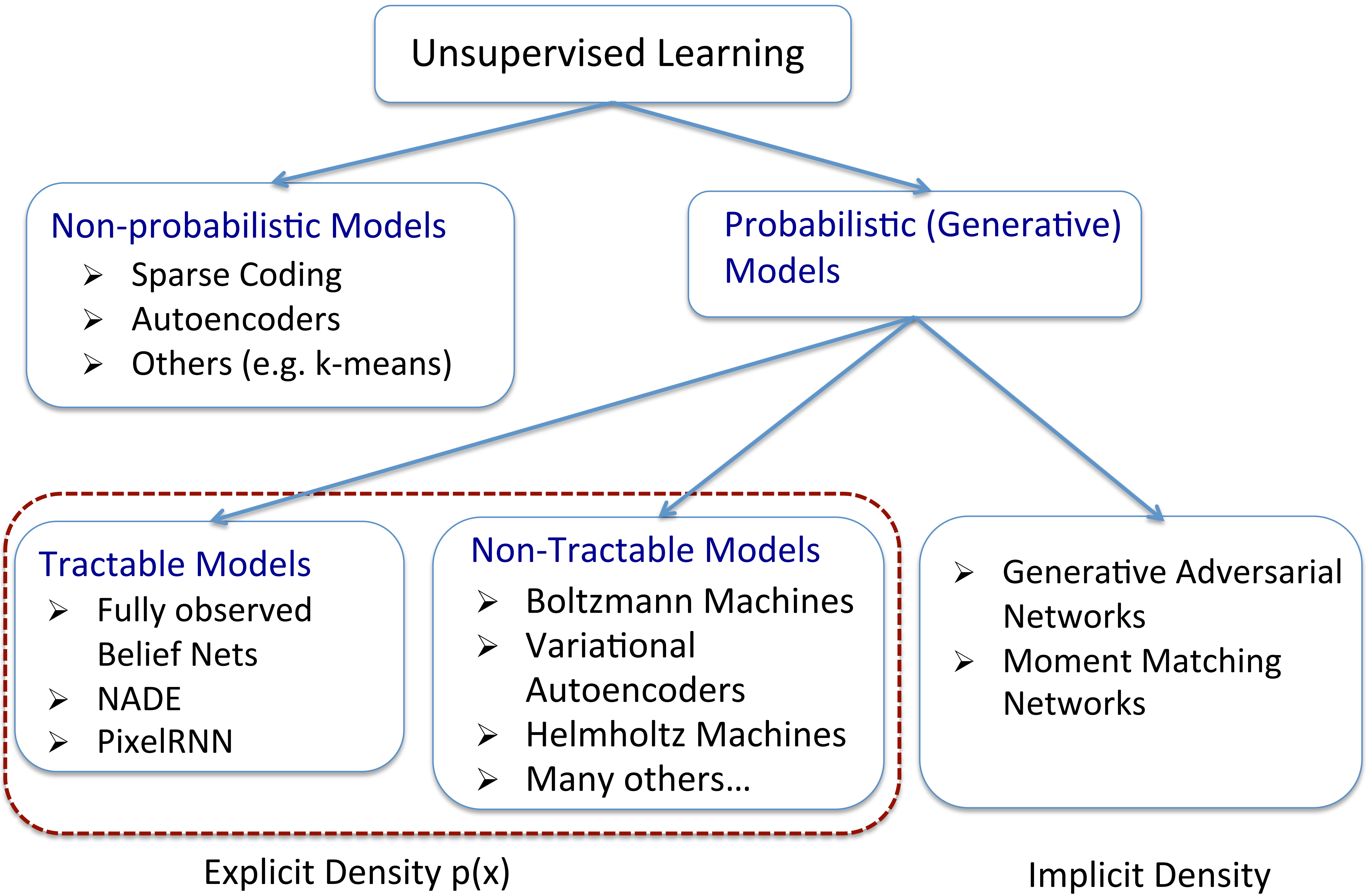
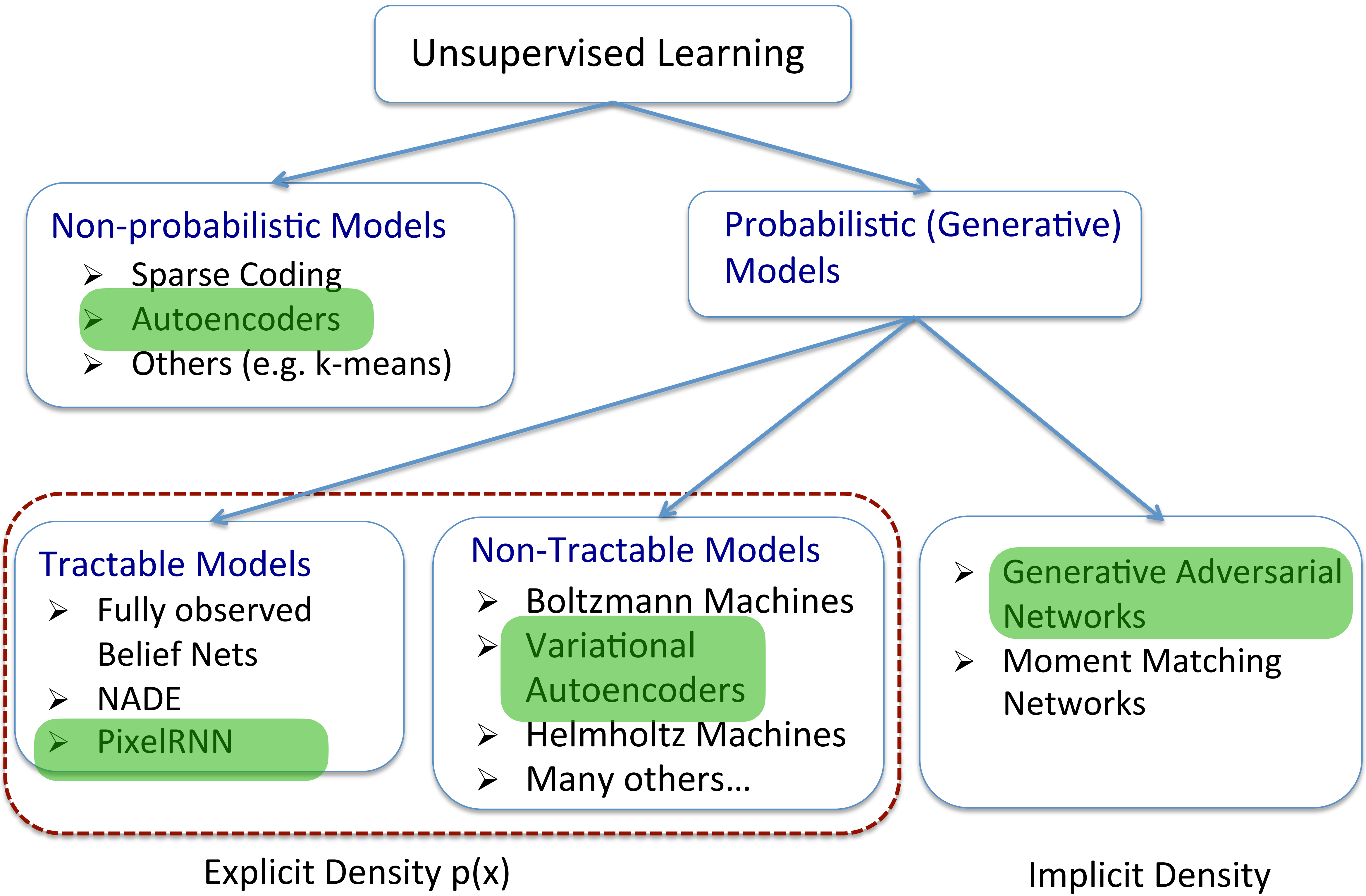
Deep Learning with Neural Networks

Unsupervised Deep Learning (I): autoencoders

Pablo Martínez Olmos, pamartin@ing.uc3m.es

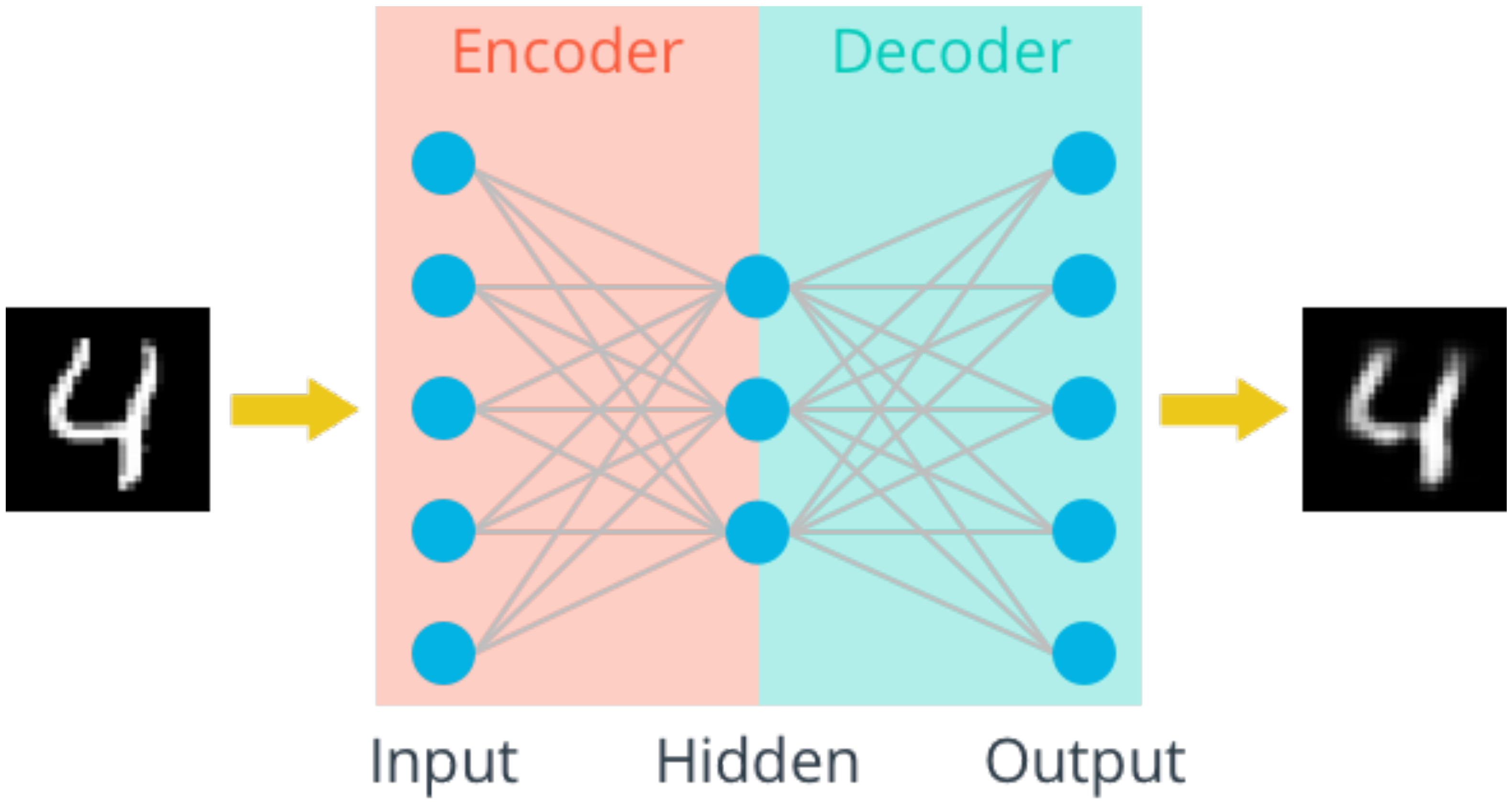

Unsupervised Learning
Non-probabilistic Models
Sparse Coding
Autoencoders
Others (e.g. k-means)
Probabilistic (Generative) Models
Tractable Models
Fully observed Belief Nets
NADE
PixelRNN
Non-Tractable Models
Boltzmann Machines
Variational Autoencoders
Helmholtz Machines
Many others…
Generative Adversarial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



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

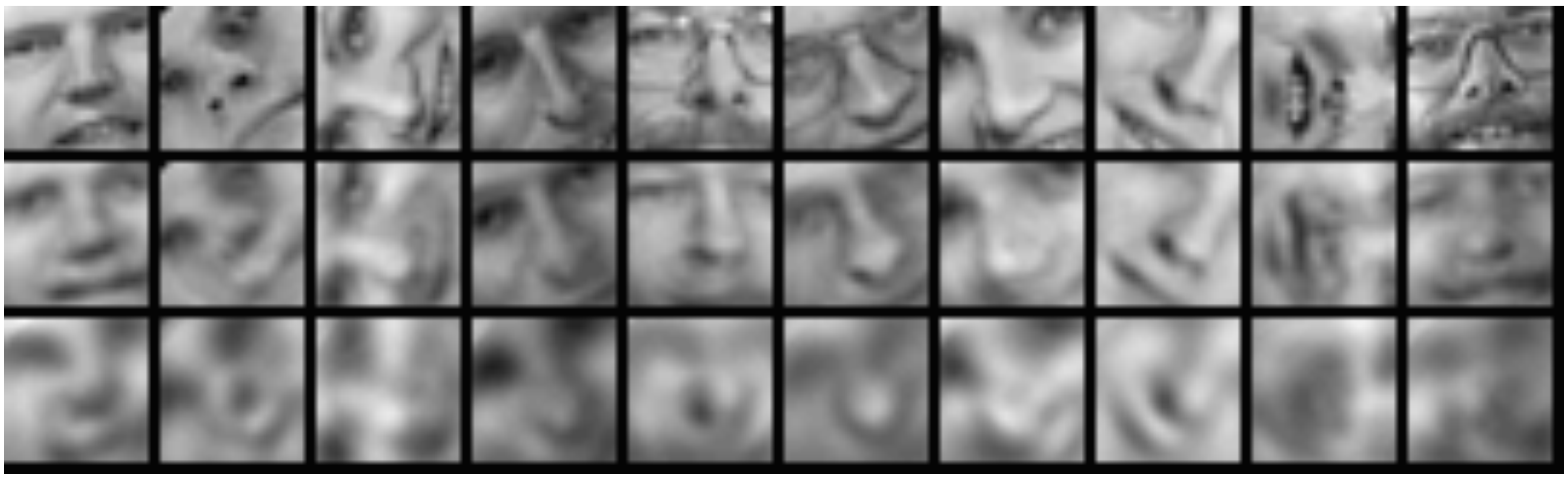
- An autoencoder is a neural network that is trained to attempt to copy its input to its output
- Internally, it has a low-dimensional hidden representation
- Two parts: encoder and decoder
- Autoencoders have been traditionally used for dimensionality reduction or feature learning



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

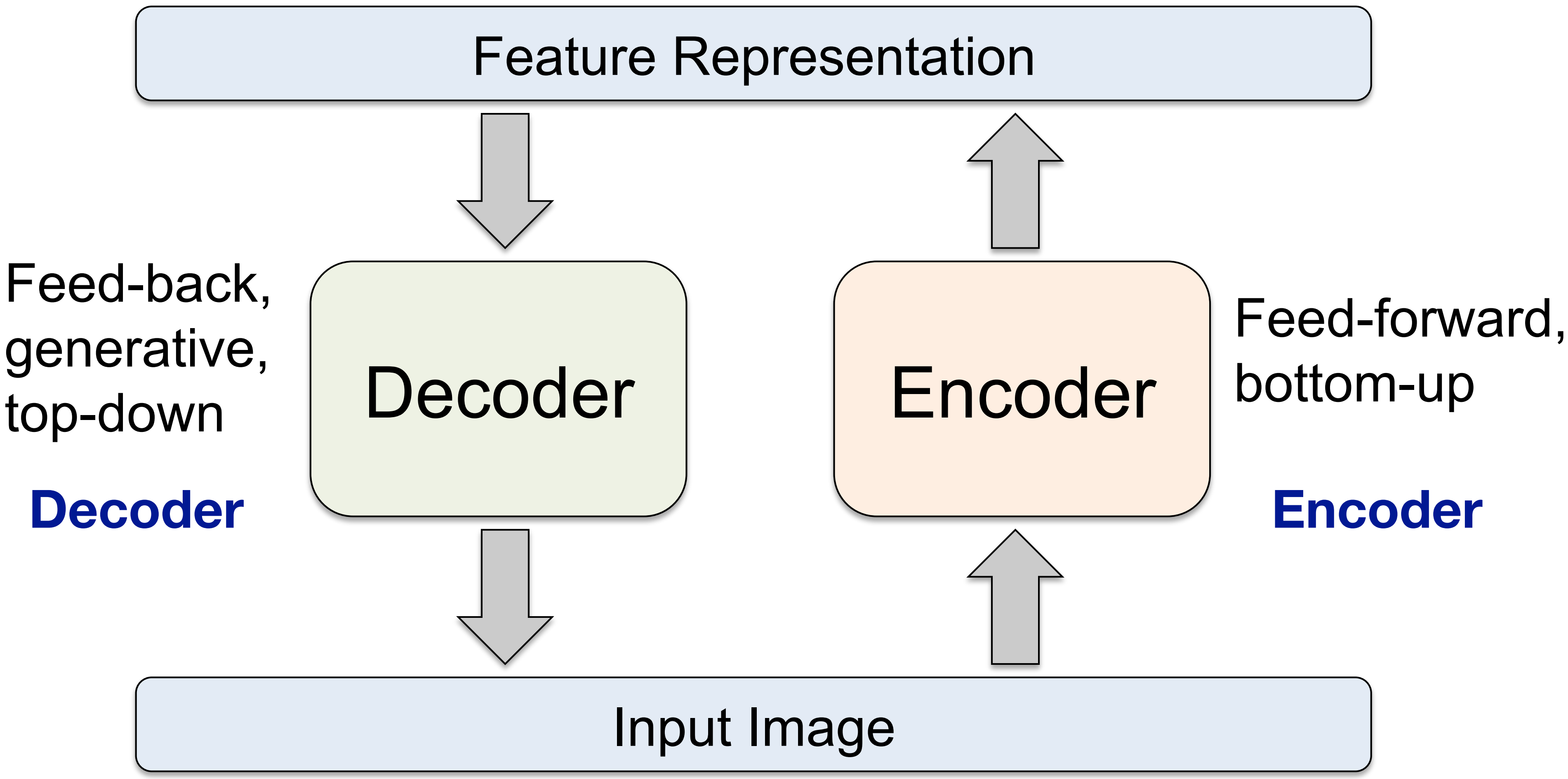
Deep Autoencoder

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

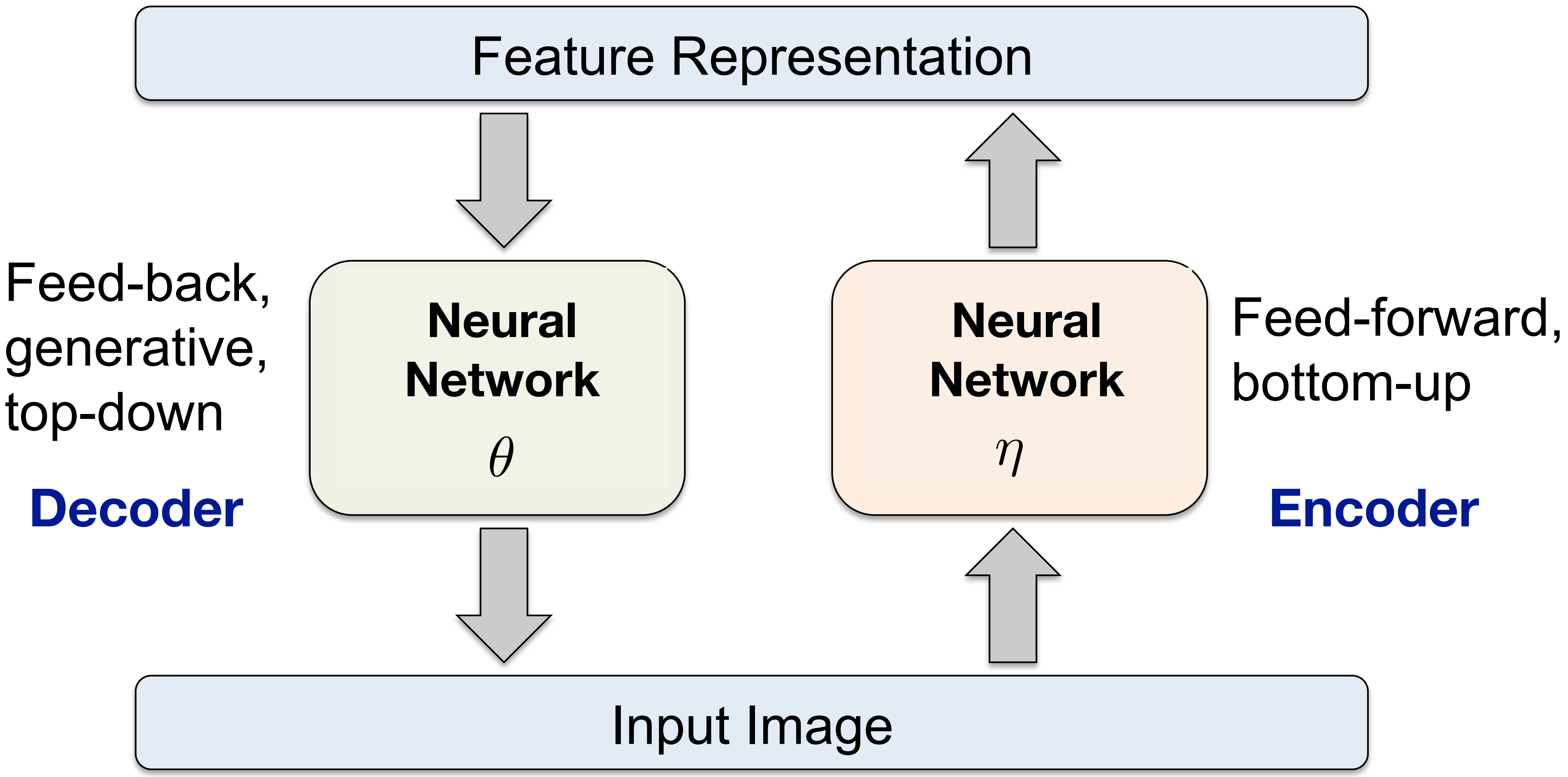


- **Top**: Random samples from the test dataset.
- **Middle**: Reconstructions by the 30-dimensional deep autoencoder.
- **Bottom**: Reconstructions by the 30-dimentinoal PCA.

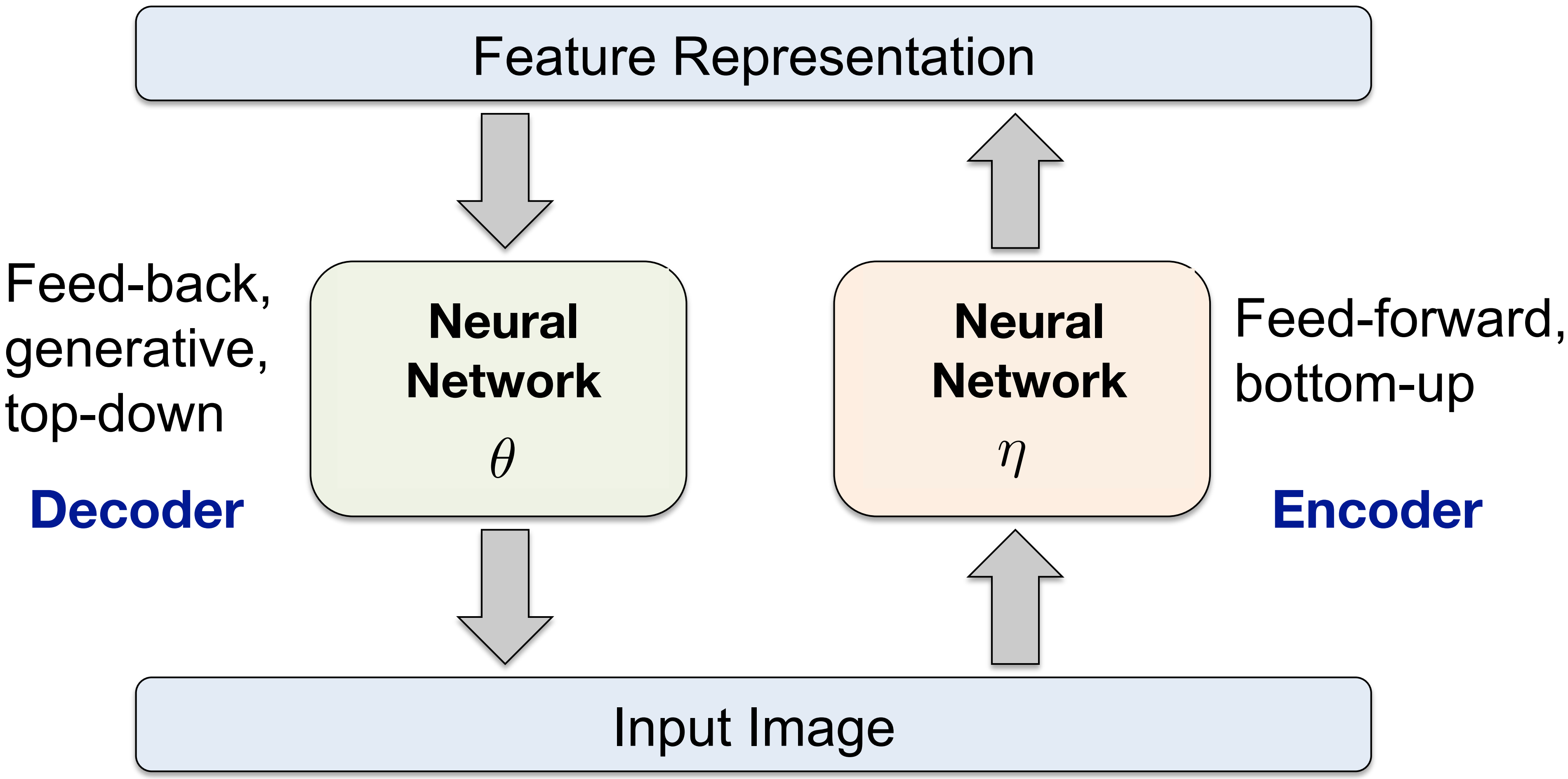
Deep Autoencoders

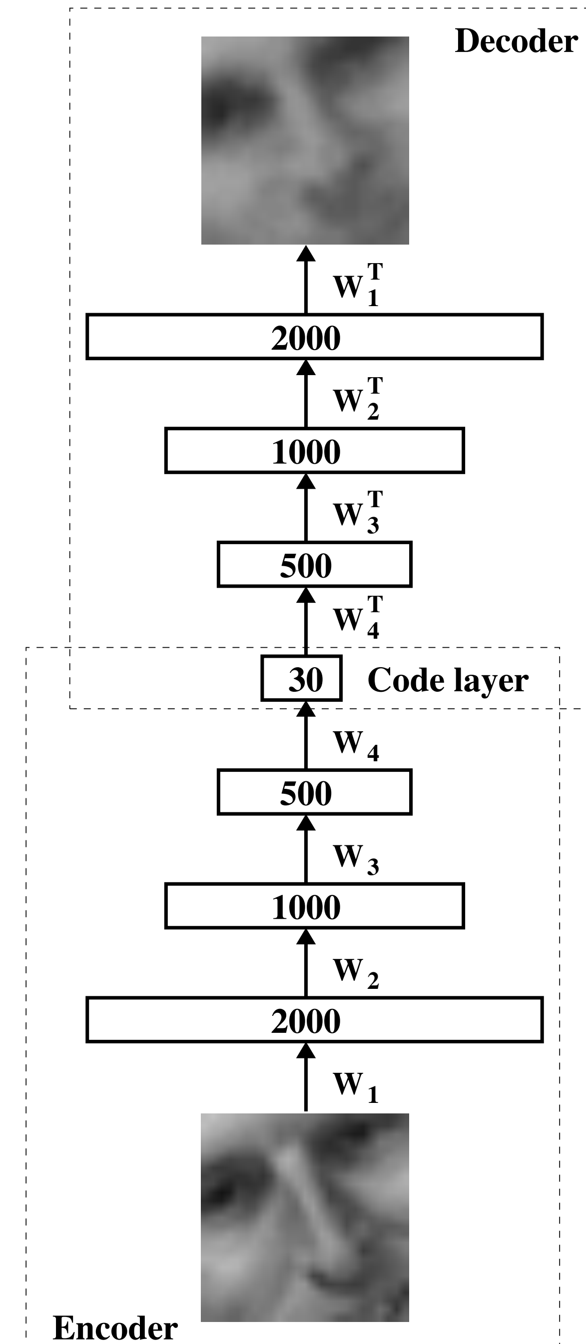


Deep Autoencoders

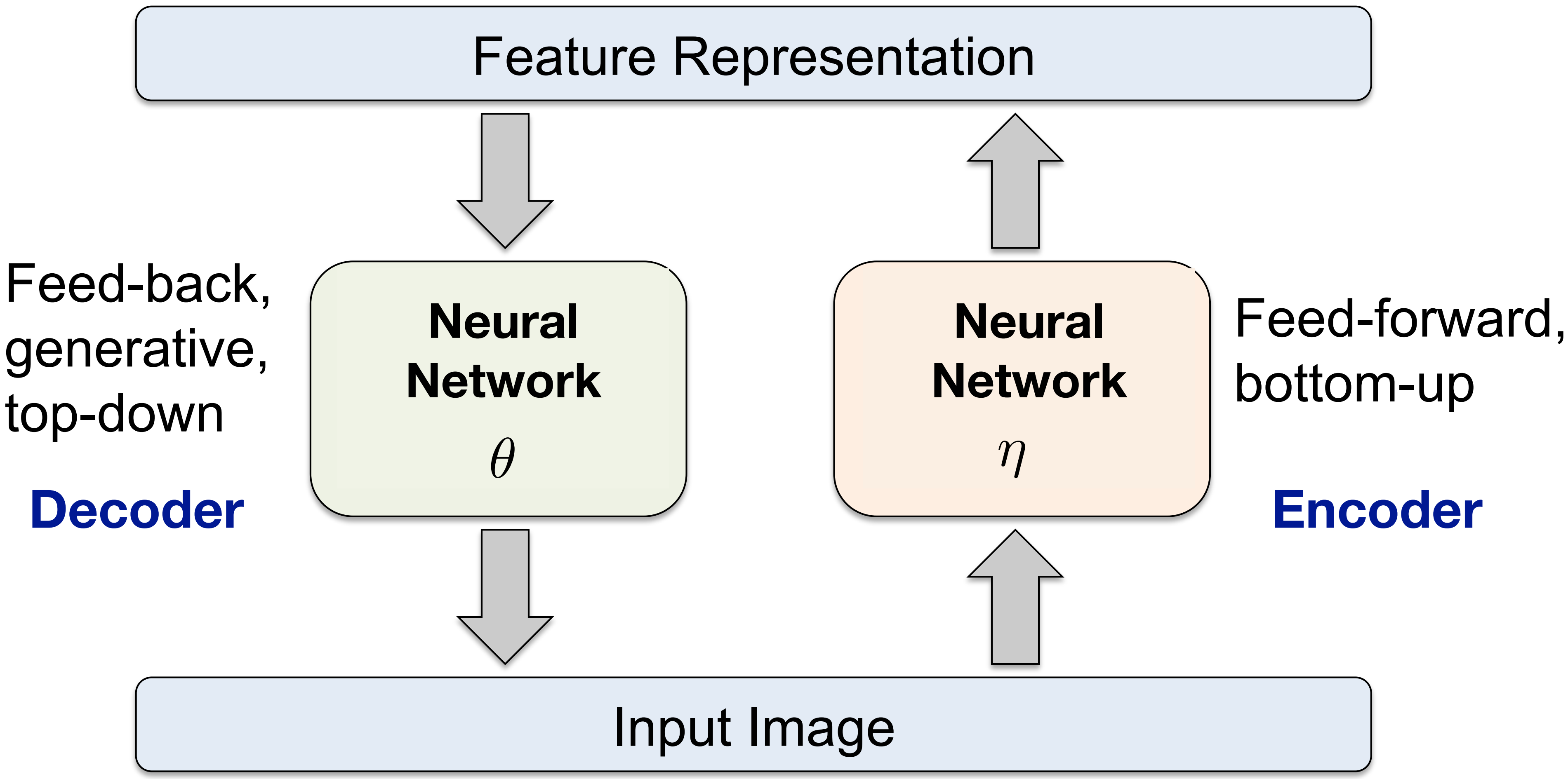


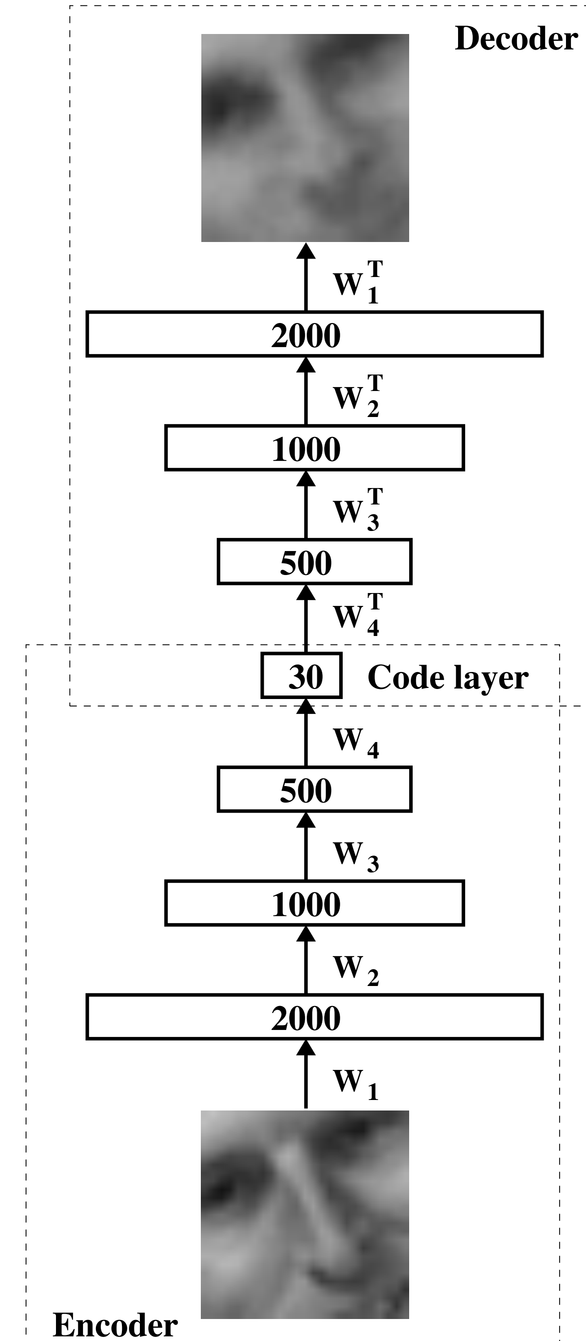
Deep Autoencoders





Deep Autoencoders

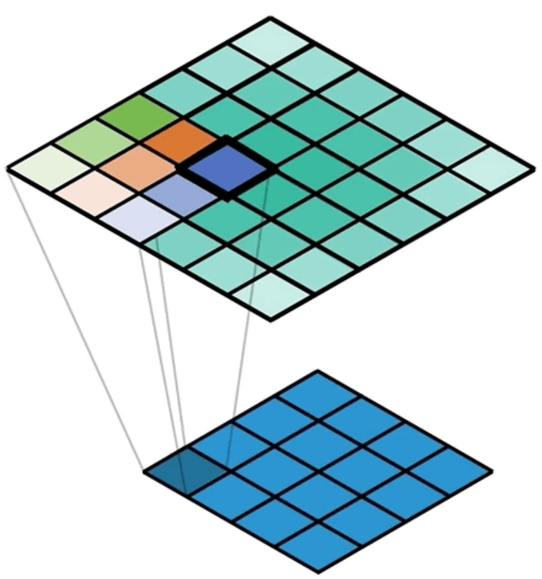


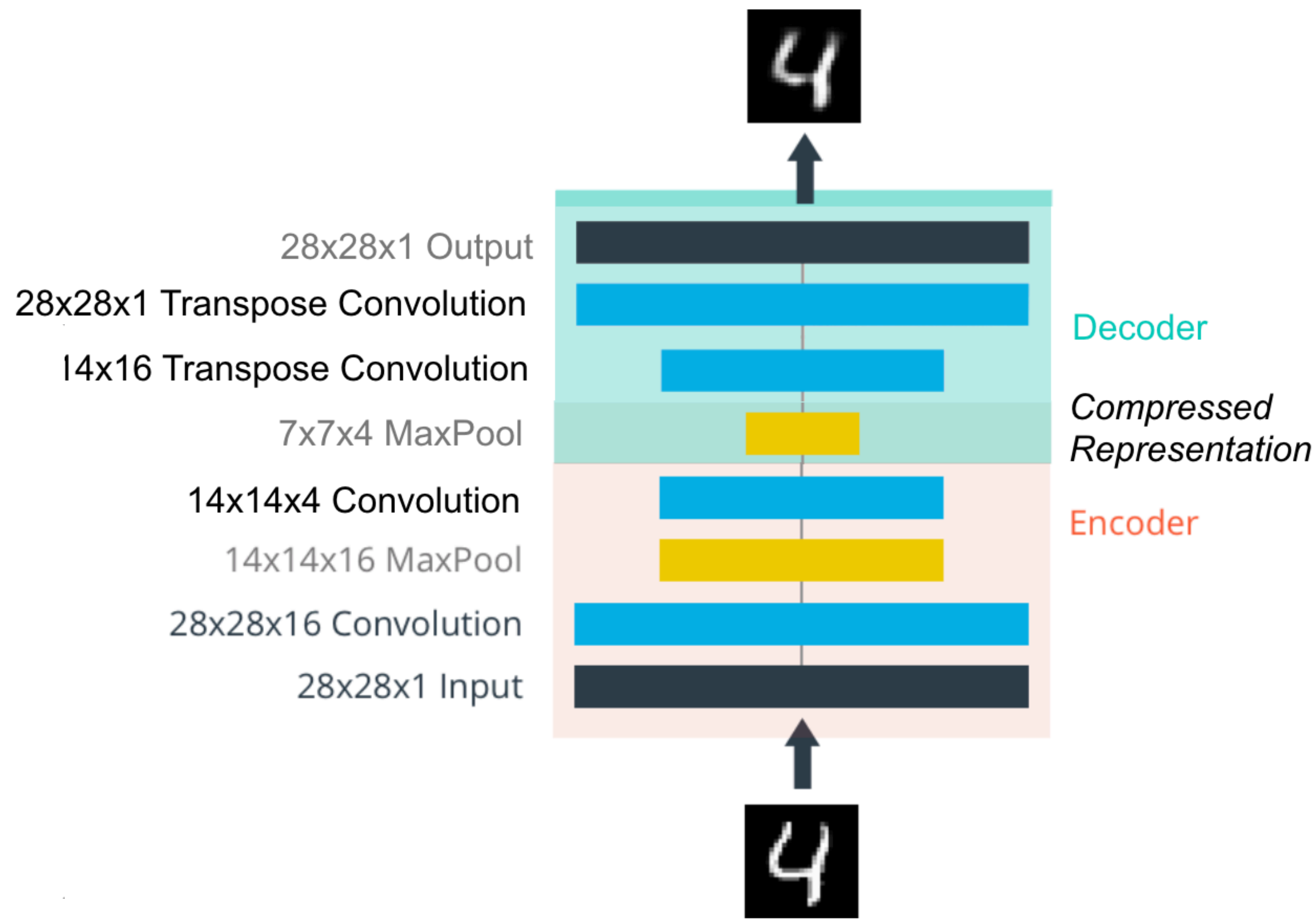


$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta (\mathbf{x}_n) \right) \right)$$

Autoencoders for images. Transpose convolution layers

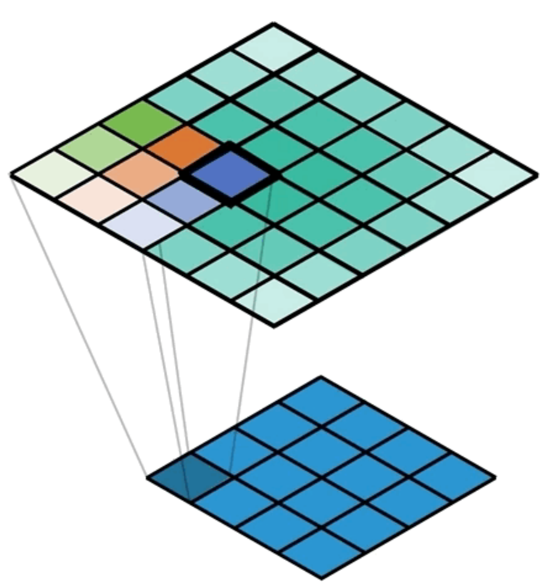
- The transposed convolution operation forms the same connectivity as the normal convolution but in the **backward direction**
- We can use it to **conduct up-sampling**
- The **weights in the transposed convolution are learnable**. So we do not need a predefined interpolation method.

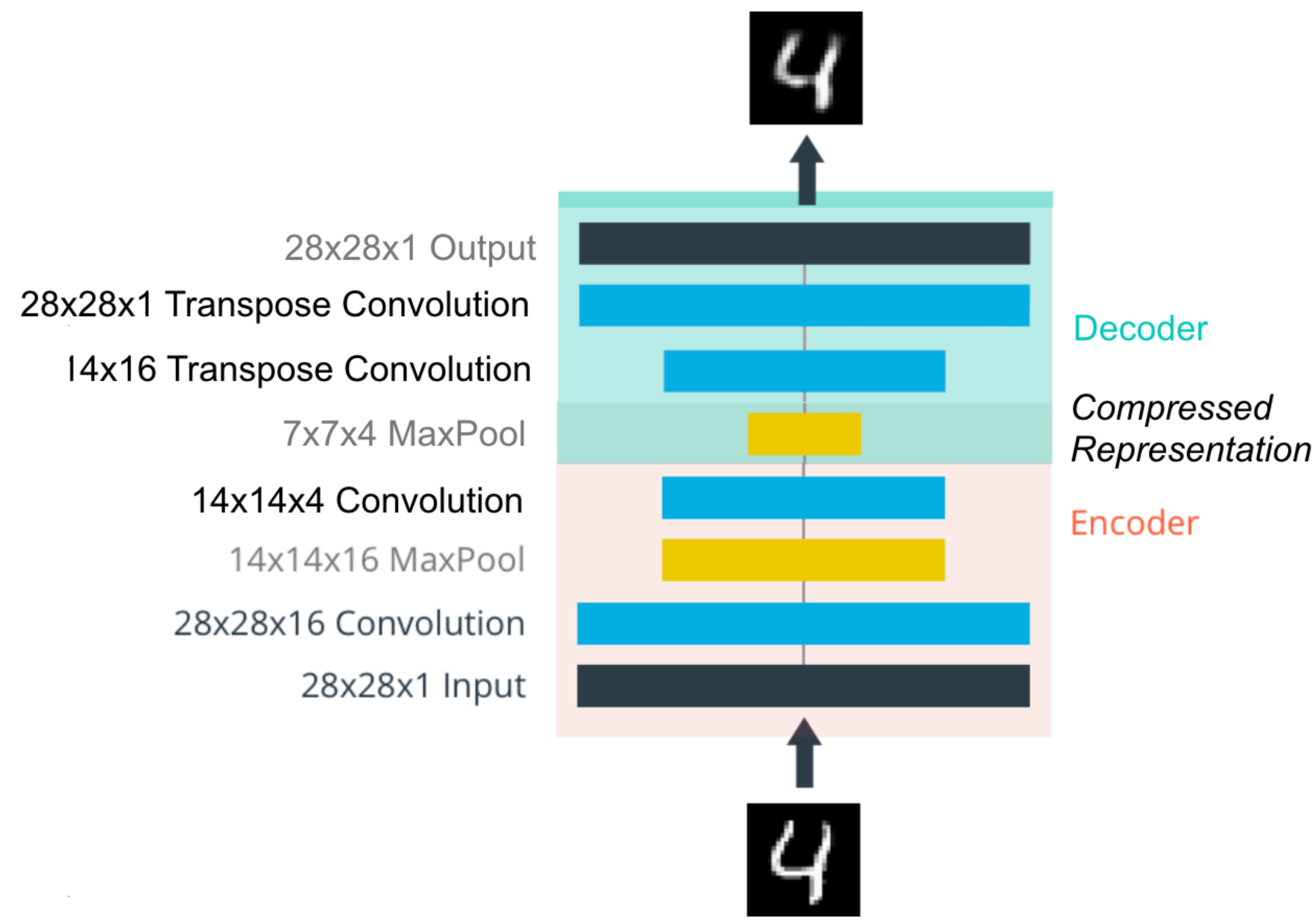




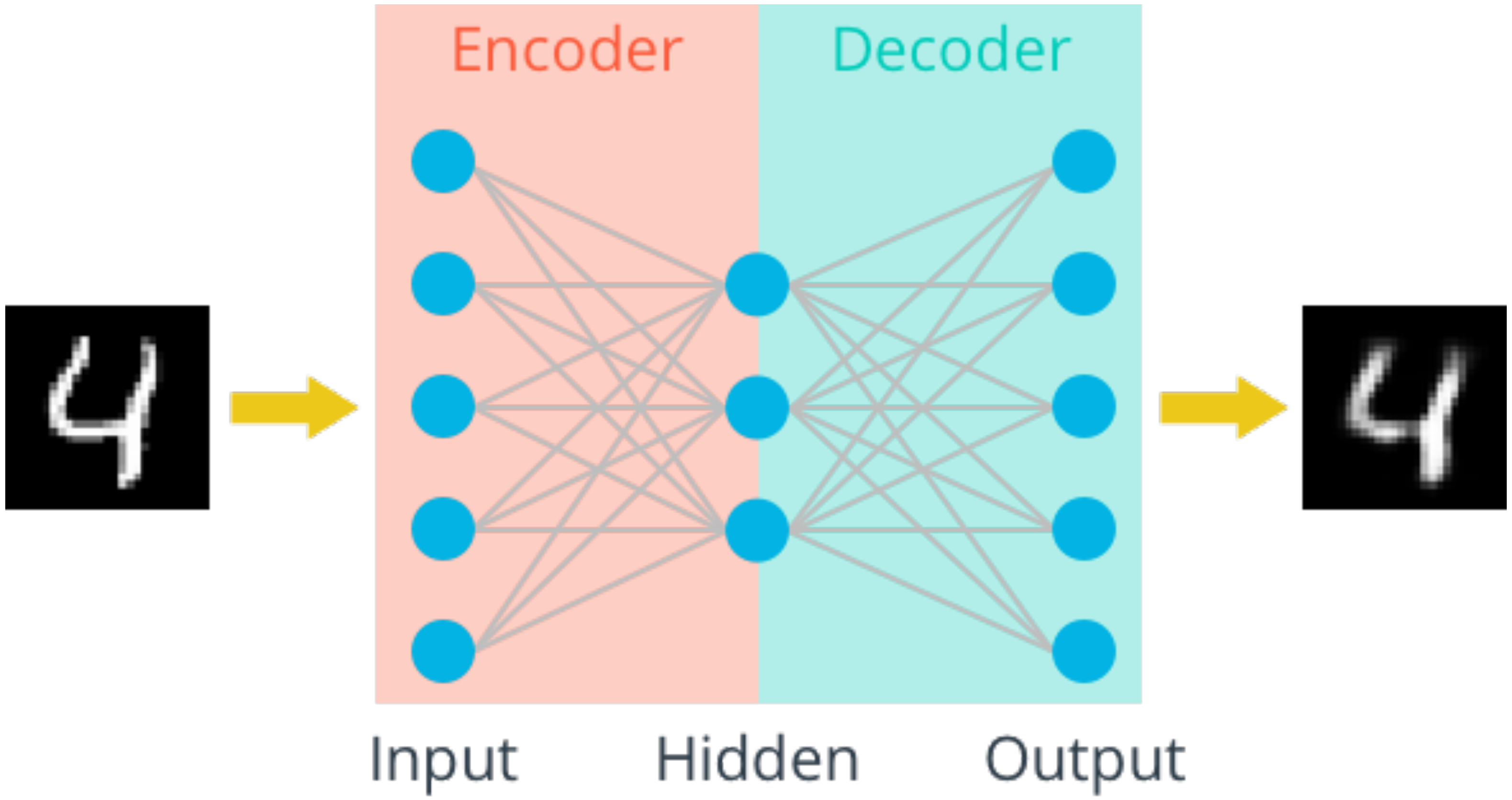
Autoencoders for images. Transpose convolution layers

- The transposed convolution operation forms the same connectivity as the normal convolution but in the **backward direction**
- We can use it to **conduct up-sampling**
- The **weights in the transposed convolution are learnable**. So we do not need a predefined interpolation method.





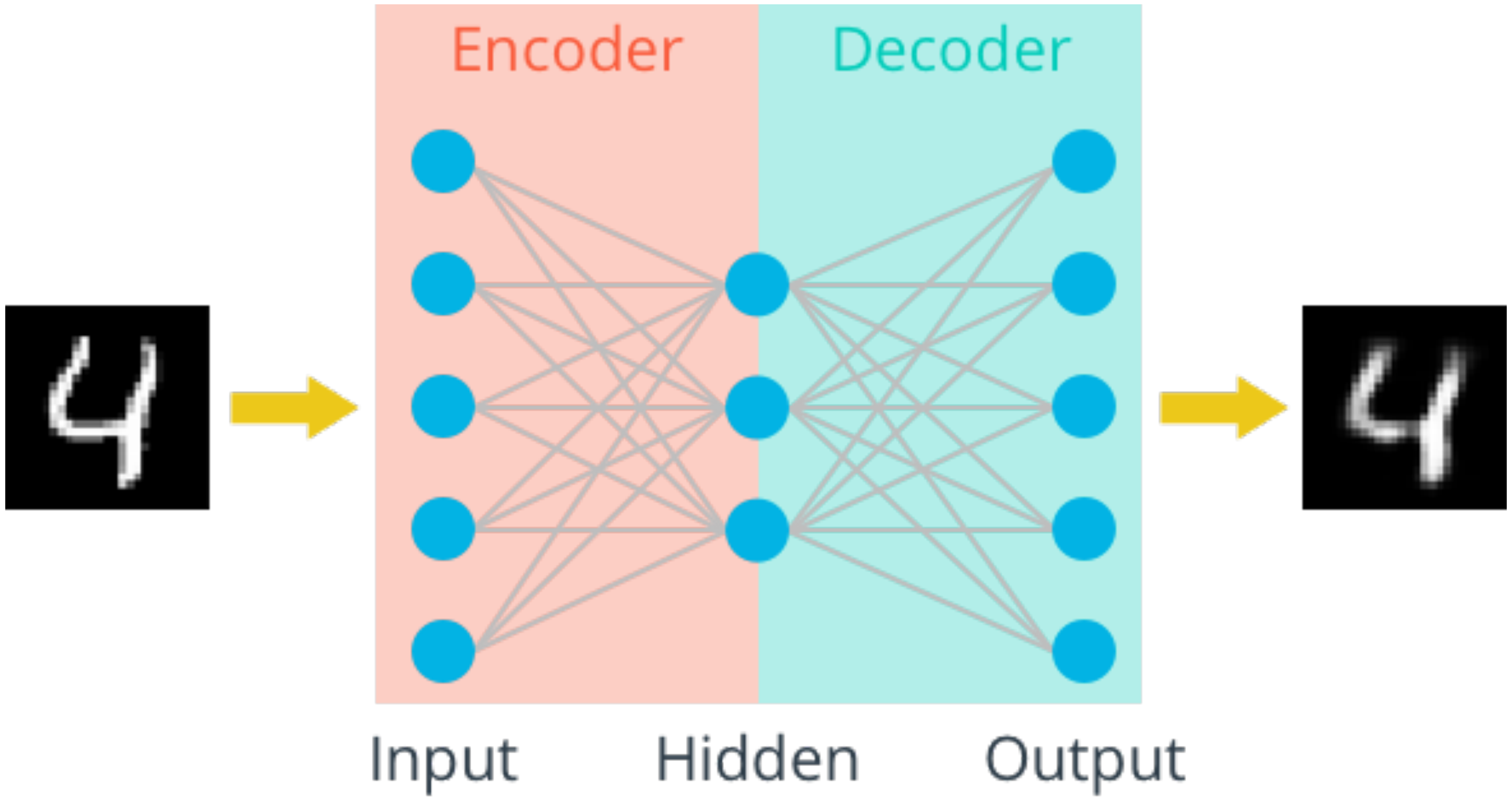
Sparse Deep Autoencoders



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

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

Sparse Deep Autoencoders



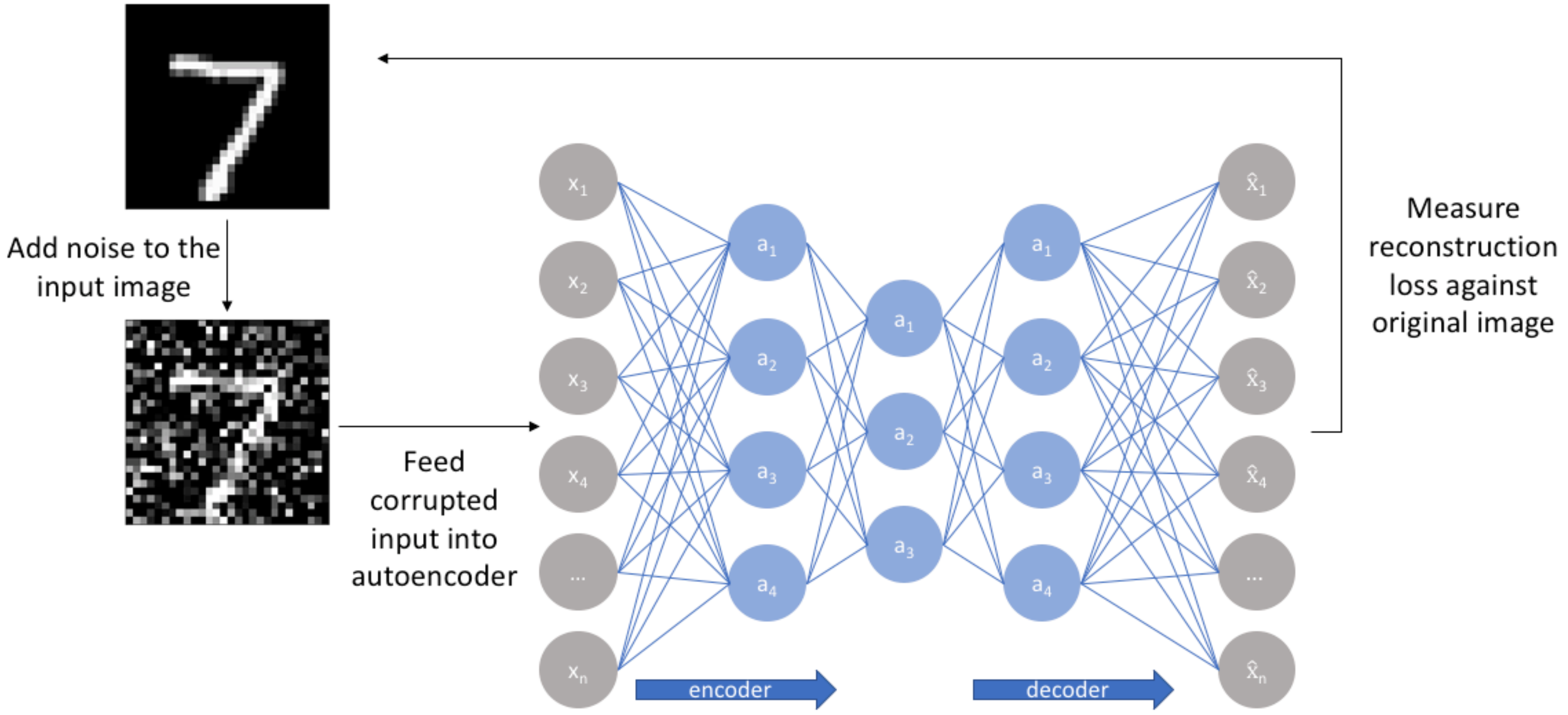
$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h})$$

Sparsity penalizer

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

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

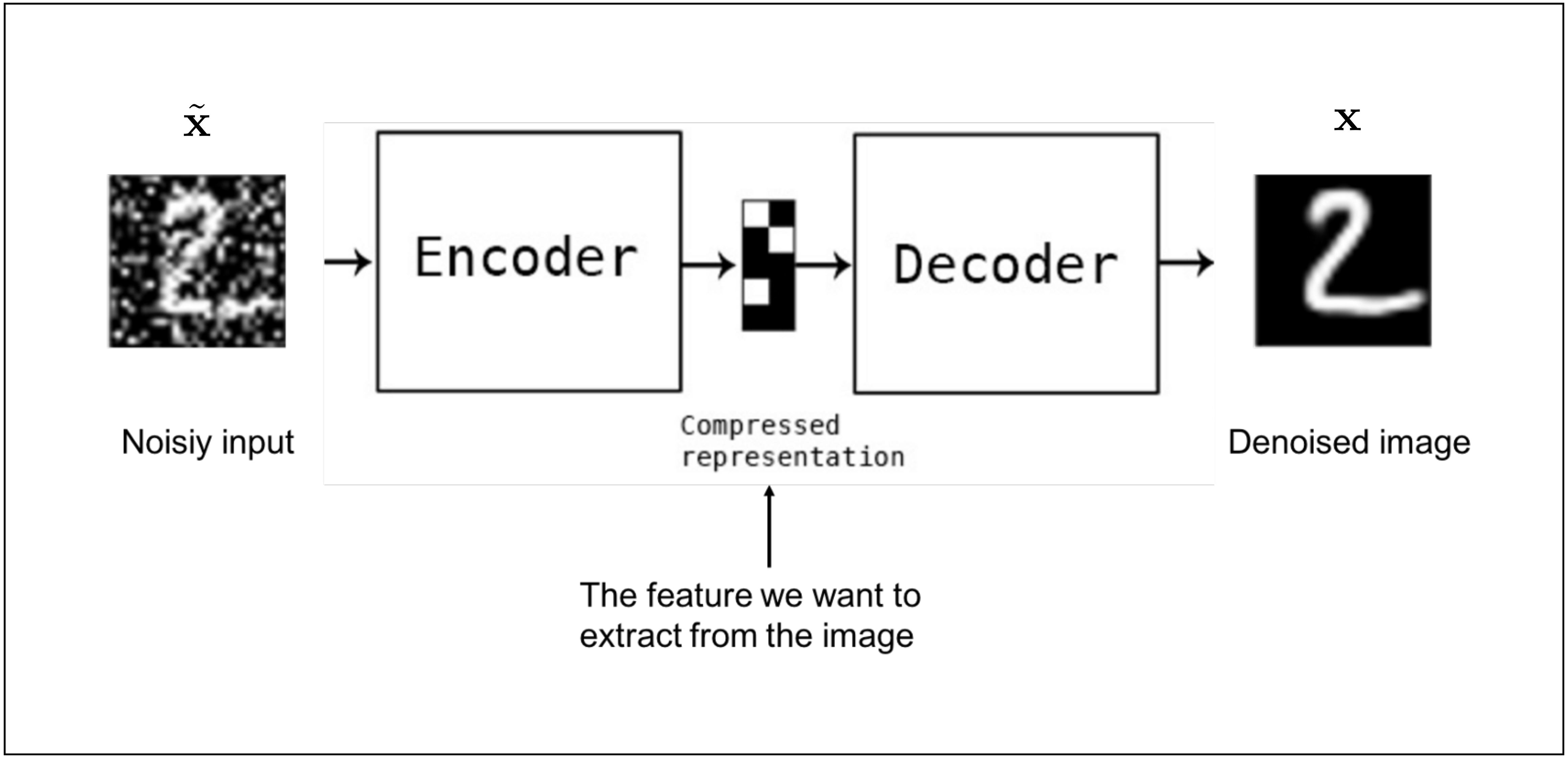
Denoising Deep Autoencoders



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



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

Image segmentation using CNNs

Image Segmentation Using Deep Learning: A Survey

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos

- Most powerful methods are based on **encoder-decoder networks**

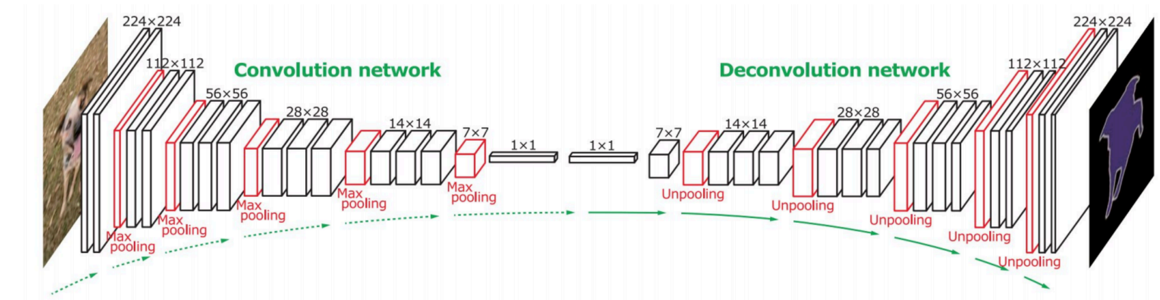


Fig. 11. Deconvolutional semantic segmentation. Following a convolution network based on the VGG 16-layer net, is a multi-layer deconvolution network to generate the accurate segmentation map. From [42].

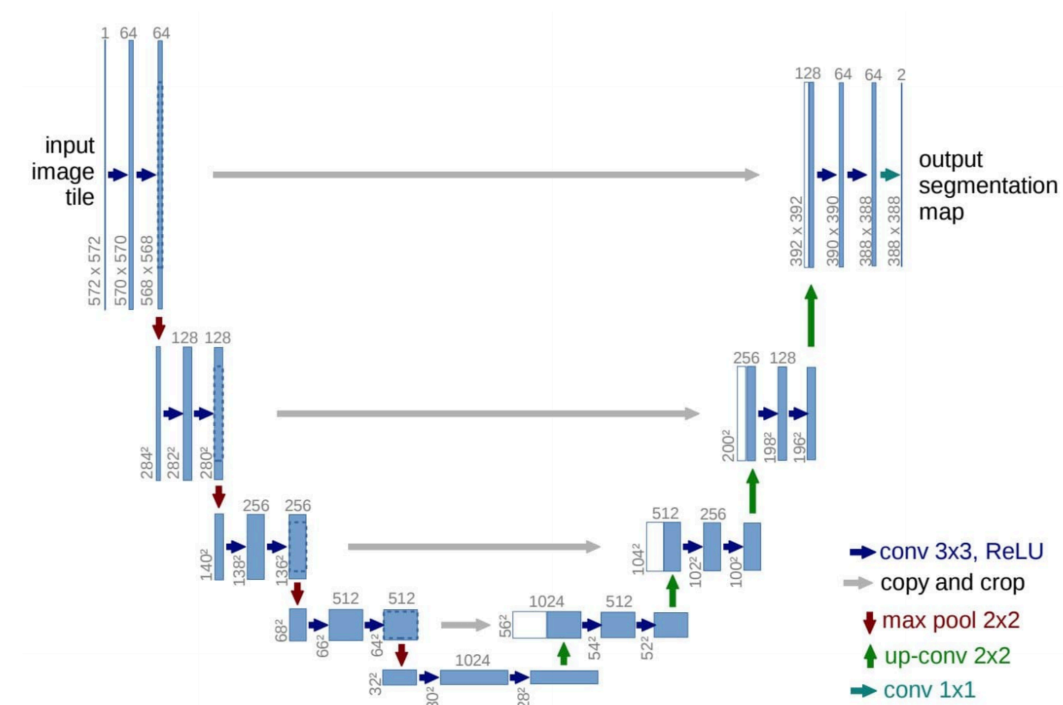


Fig. 14. The U-net model. The blue boxes denote feature map blocks with their indicated shapes. From [49].