

Applied Machine Learning

Autoencoders

Autoencoders

- Structure of autoencoders
- deconvolutional layers
- unpooling layers
- sigmoid layers
- cost function for training autoencoders

Encoders and Decoders

- Encoder
 - Input: high-dimensional data item
 - Output: low-dimensional representation of data item that preserves relevant information
 - CNNs layers as encoders
 - Map pixels to classes
- Decoder
 - Input: low-dimensional representation of data item
 - Output: original high-dimensional data item
 - NN layers as decoders
 - learn reverse mapping to predict original item

Autoencoders

- Pair of encoder and decoder that are trained together

- Encoder: \mathcal{E}

- maps item \mathbf{x}_i into code \mathbf{z}

- $\mathbf{z} = \mathcal{E}(\mathbf{x}_i)$

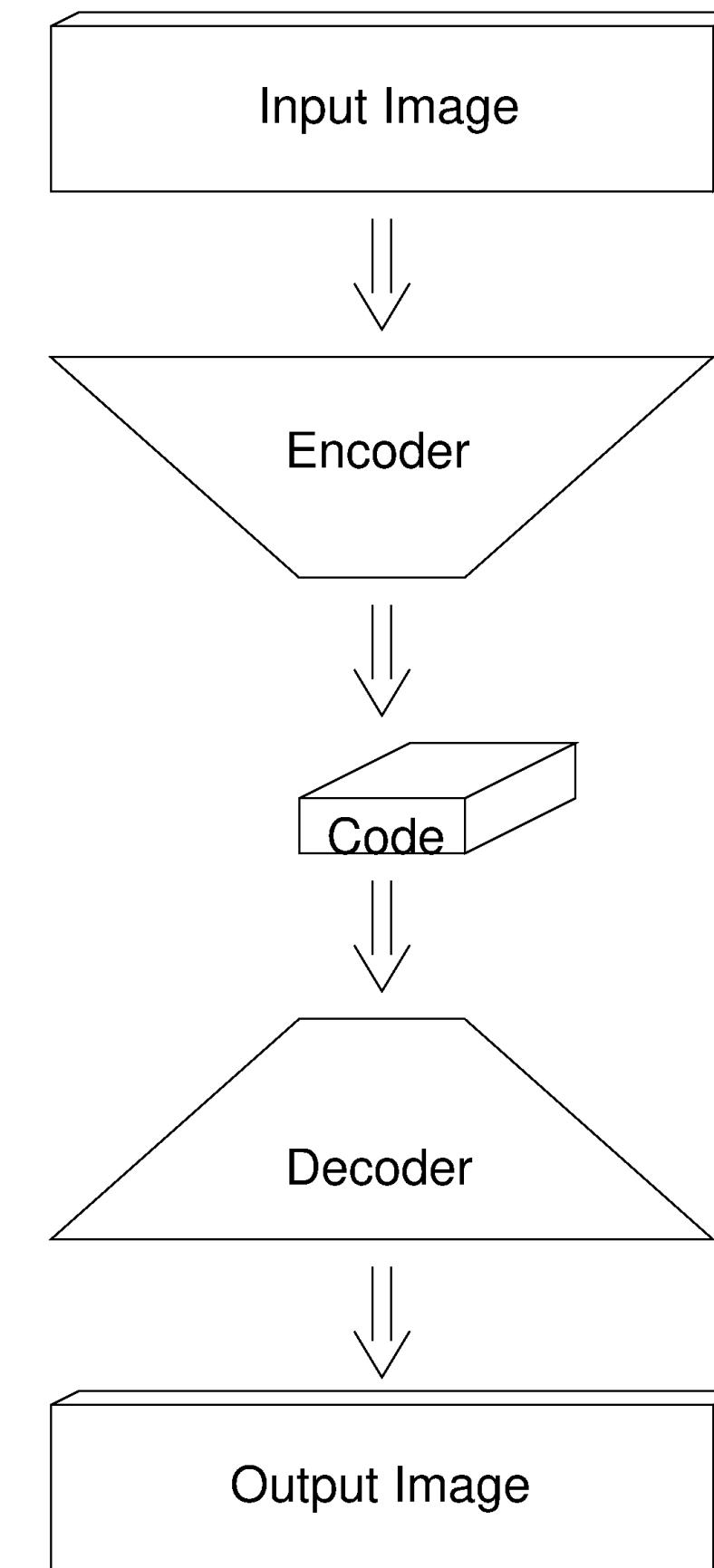
- Decoder: \mathcal{D}

- reconstructs item \mathbf{x}_o from code \mathbf{z}

- $\mathbf{x}_o = \mathcal{D}(\mathbf{z})$

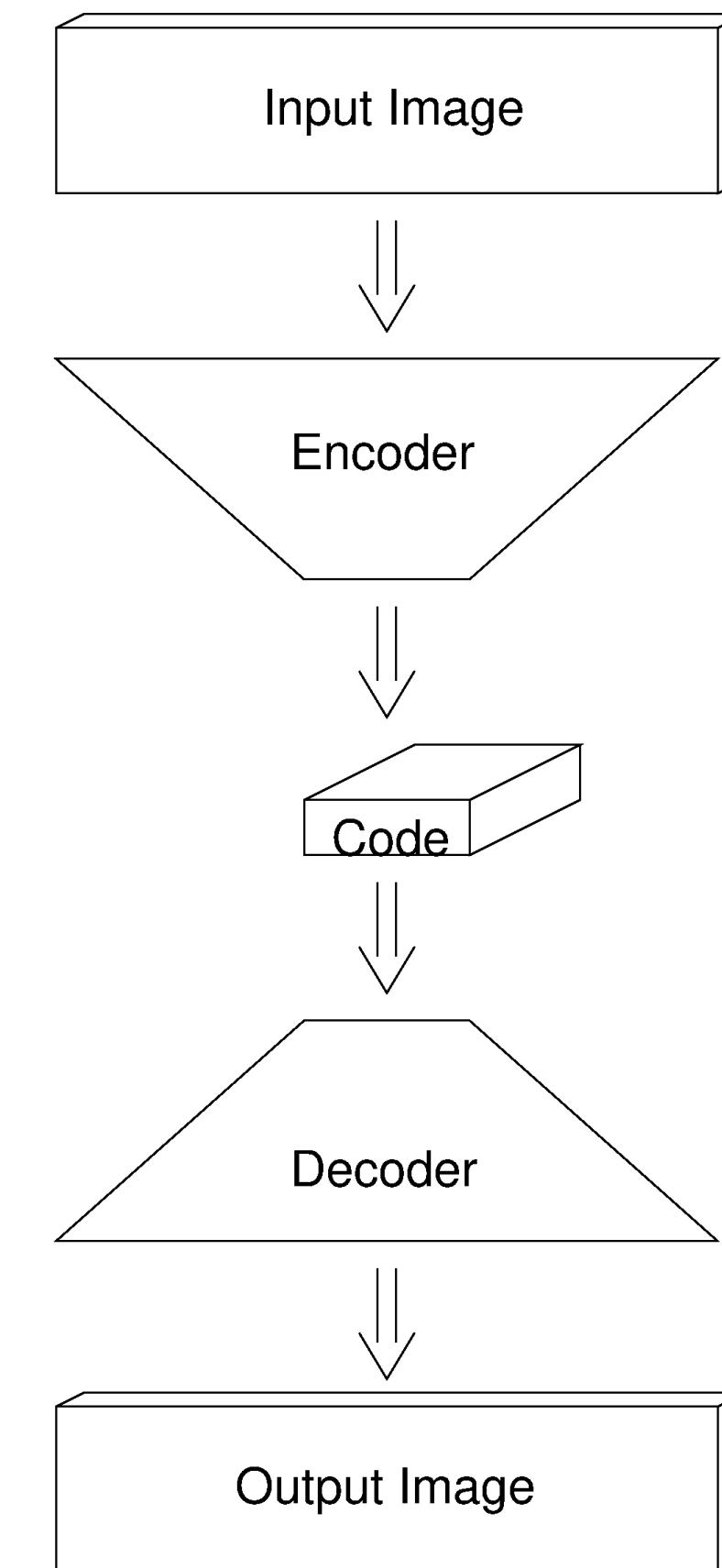
- Training goal: output similar to input

- $\mathbf{x}_o \approx \mathbf{x}_i$



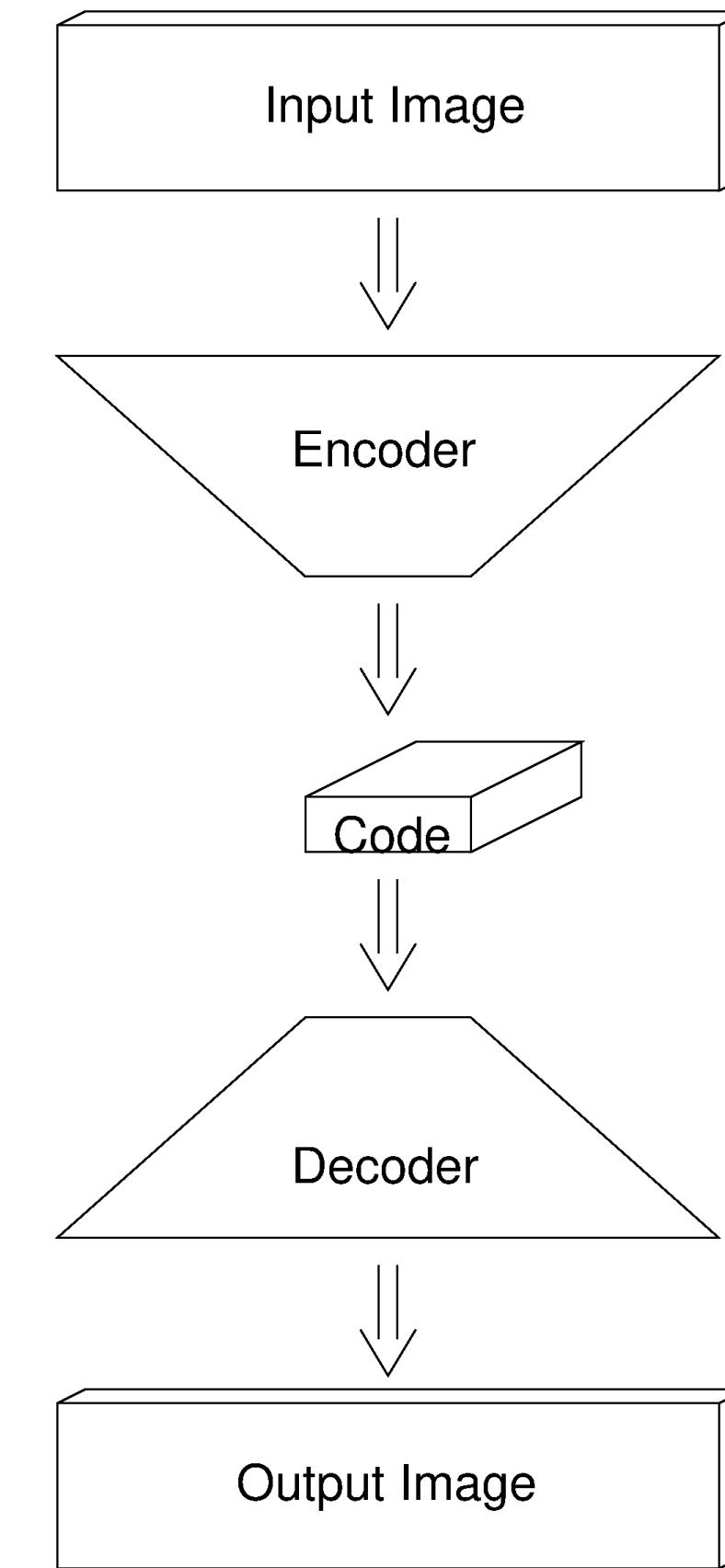
Autoencoders

- Encoder: \mathcal{E} $\mathbf{z} = \mathcal{E}(\mathbf{x}_i)$
- Decoder: \mathcal{D} $\mathbf{x}_o = \mathcal{D}(\mathbf{z})$
- Training goal: $\mathbf{x}_o \approx \mathbf{x}_i$
- Some Applications:
 - Unsupervised feature learning
 - \mathbf{z} : feature set, \mathbf{x}_o : validation
 - Clustering
 - \mathbf{z} : cluster distribution, \mathbf{x}_o : validation
 - Image Generation
 - \mathbf{z} : image distributions, \mathbf{x}_o : reconstructed images



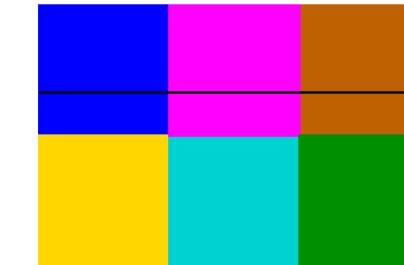
Autoencoder for Images

- Input: Image I_i with dimensions $i_u \times i_v \times 1$
- Encoder layers
 - convolutional + pooling layers
 - stride length $s > 1$
 - parameters: θ_e
- Output code: Feature map $Z_i = \mathcal{E}(I_i, \theta_e)$
 - Dimensions $z_u \times z_v \times l$
 - Usually smaller than input $z_u \times z_v \ll i_u \times i_v$
- Decoder layers mirror structure of the encoder layers
 - deconvolutional layer mirrors convolutional layer
 - unpooling layer mirrors pooling layer



Deconvolutional Layer

- Deconvolutional or transposed convolutional layer
 - Complementary to convolutional layer
 - Increases spatial dimensionality of input
 - input dimensions: $u \times v \times 1$
- Output feature map
 - Stride length: s
 - dimensions:
 $[u + u * (s - 1)] \times [v + v * (s - 1)] \times 1$
 - Kernel dimensions: $[2 * k + 1] \times [2 * k + 1]$



Unpooling Layer

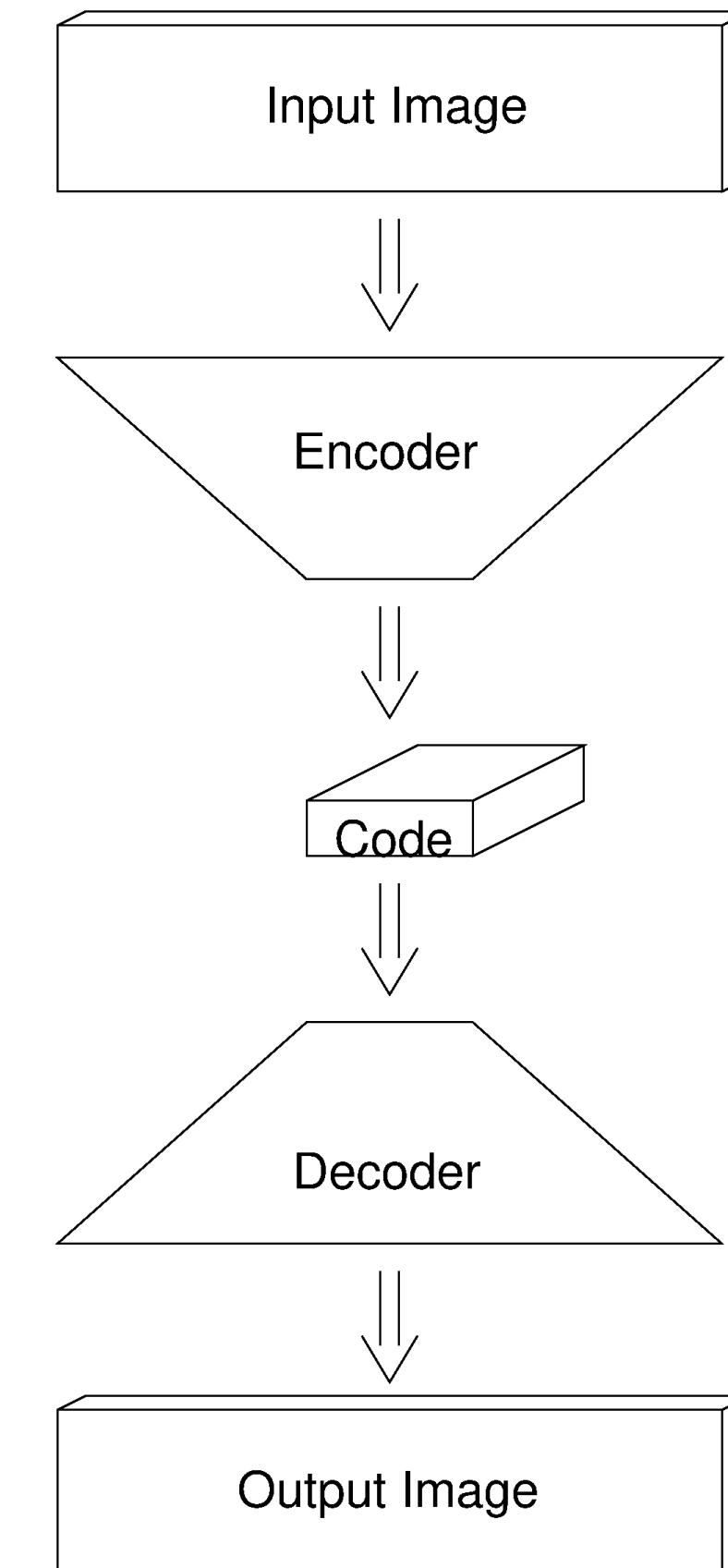
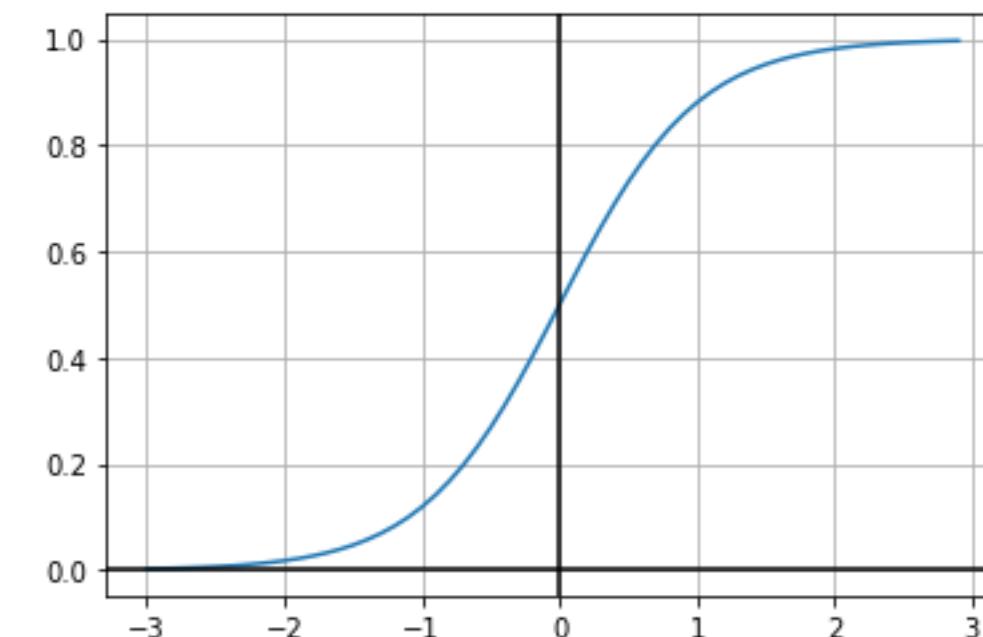
- Unpooling Layer
 - Complementary to function applied at the pooling layer
 - same patch size p as in the mirrored pooling layer
- Output feature map with one patch per input pixel initialized with 0s
 - average unpooling
 - copy input pixel value to all pixels in the output patch
 - maximum pooling:
 - Modification to mirrored pooling layer
 - max-pooling map
 - map with location of source max pixel
 - copy input pixel value to output patch at the same location of max pixel at mirrored pooling layer

Avg.	
2	0.5
1.5	0

2	2	0.5	0.5
2	2	0.5	0.5
1.5	1.5	0	0
1.5	1.5	0	0

Sigmoid Layer

- Output decoder layer
 - Image
 - non-linear
 - sigmoid: $s(x) = \frac{1}{1 + e^{-x}}$
 - output range: [0-1]
 - can be mapped to [0-255]
 - if needed, preprocess inputs to the encoder to be in range [0-1]



Training Autoencoder

- Input: image

$$I_i$$

- Encoder

- parameters θ_e

- Output: Feature map $Z_i = \mathcal{E}(I_i, \theta_e)$

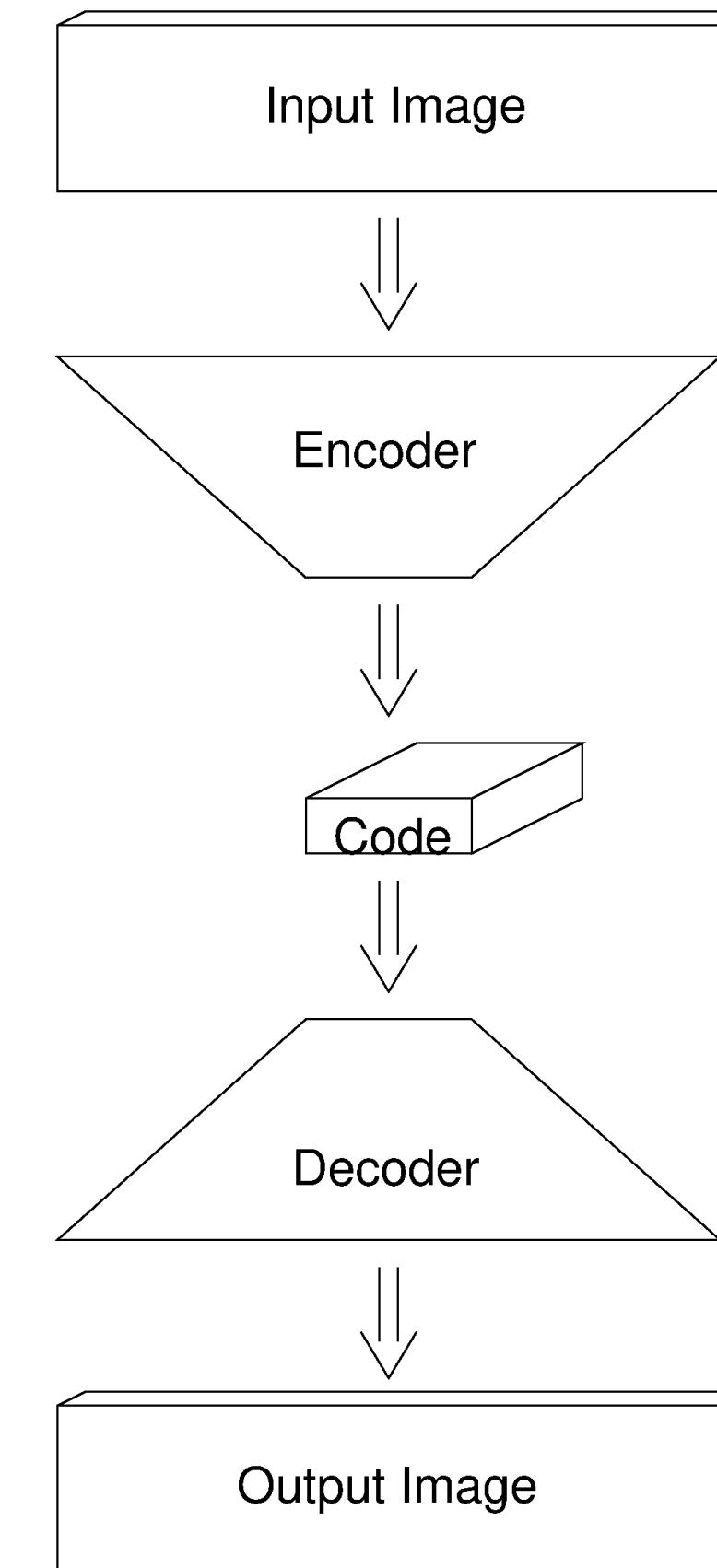
- Decoder

- parameters θ_d

- Output: image $\mathcal{D}(Z_i, \theta_d)$

- Cost function: $\|\mathcal{D}(Z_i, \theta_d) - I_i\|^2$

- susceptible to small changes in images in training set



Autoencoders

- Structure of autoencoders
- deconvolutional layers
- unpooling layers
- sigmoid layers
- cost function for training autoencoders

Applied Machine Learning

Autoencoders