Computer Vision – TP13
Advanced Deep Learning Topics

Miguel Coimbra, Hélder Oliveira

Outline

- Autoencoders
- Deep learning for segmentation

Outline

- Autoencoders
- Deep learning for segmentation

Supervised vs. Unsupervised

- Supervised learning
  - We have access to a set of training data for which we know the correct class/answer
  - Training data: $\{(x_i, y_i)\}_{i=1}^{N}$
  - x_i: data (e.g., image)
  - y_i: label

- Examples
  - Image classification
  - Image segmentation
  - Object detection
  - Etc.

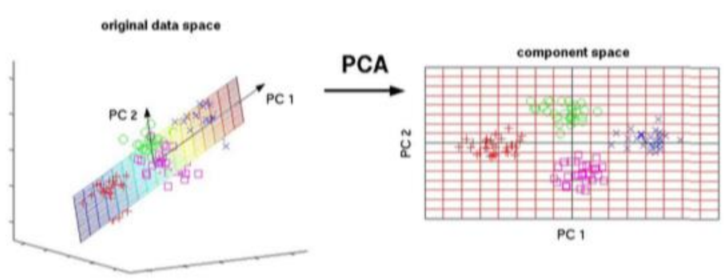
Object Detection

Semantic Segmentation

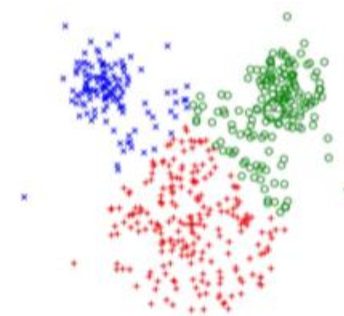
Supervised vs. Unsupervised

- Unsupervised learning
  - Discover hidden structures in the data
  - Training data: $\{x_i\}_{i=1}^{N}$
  - x_i: only data (e.g., image), no label!

- Examples
  - Clustering
  - Dimensionality reduction
  - Generative models
  - Etc.



3-d → 2-d



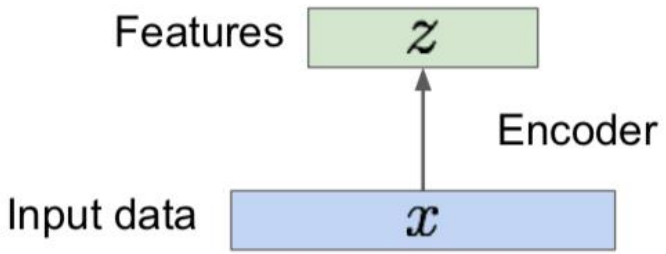
K-means clustering

Autoencoders

- Objective
  - Find representative features of the data
- Unsupervised learning
  - No data labels required
- Simple idea
  - Learn a representation of the data and try to recover the original data from that!

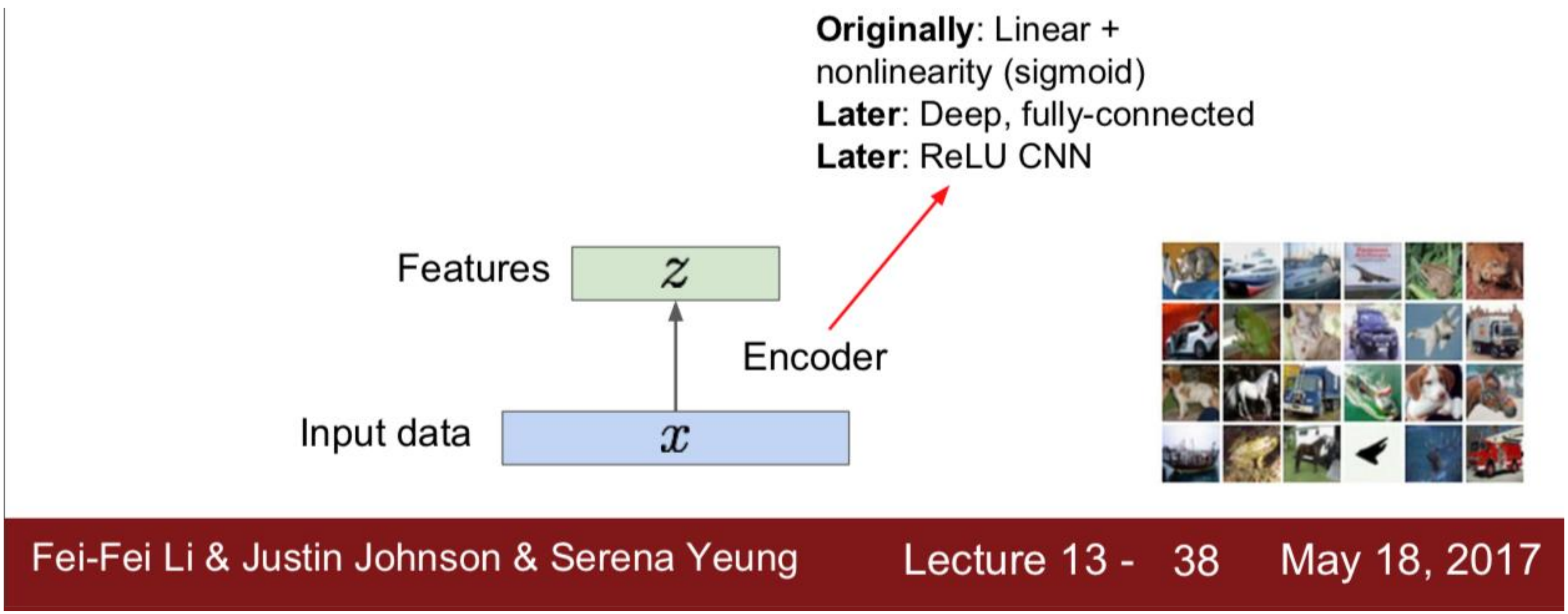
Autoencoders

- Representative features



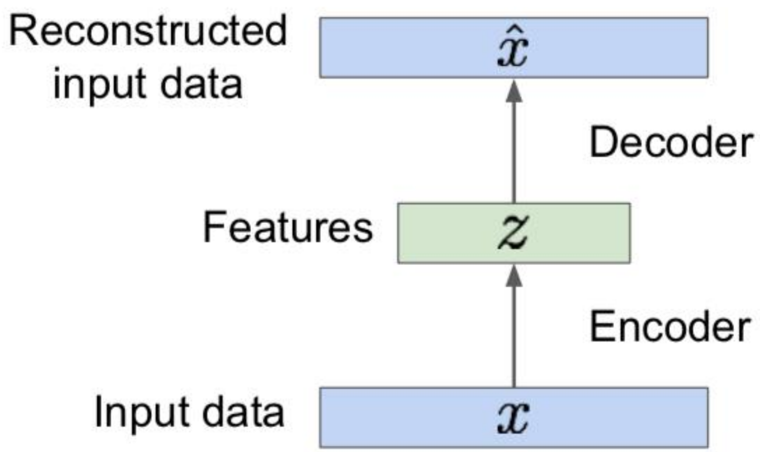
Autoencoders

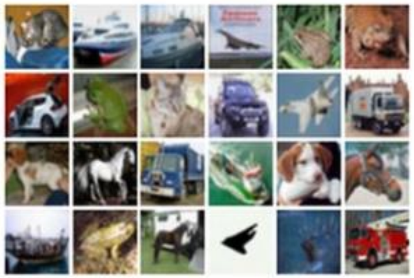
- Representative features



Autoencoders

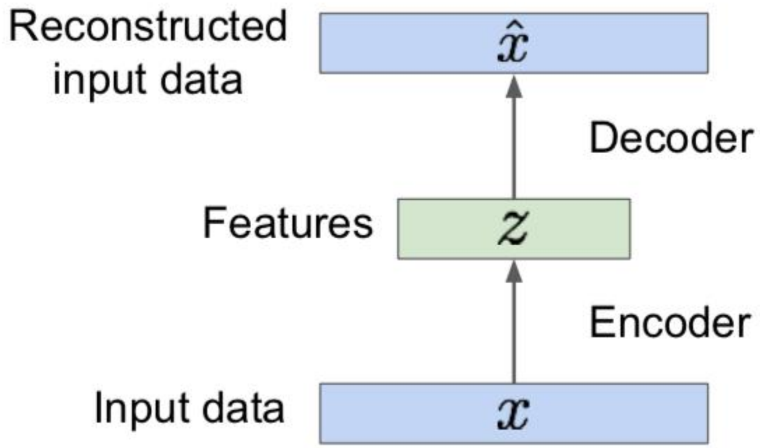
- Reconstruction



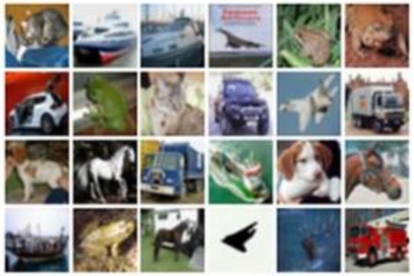


Autoencoders

- Reconstruction

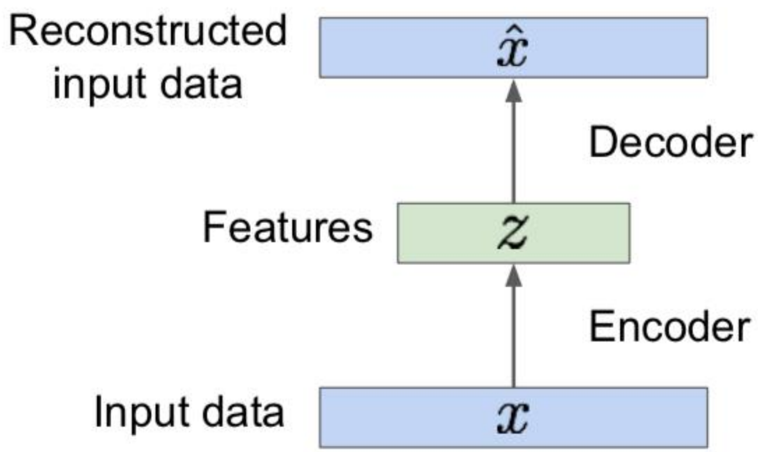


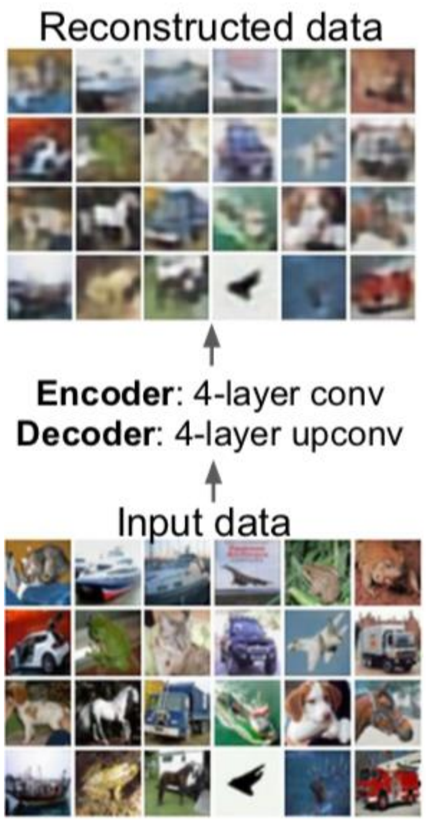
Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN (upconv)



Autoencoders

- Reconstruction

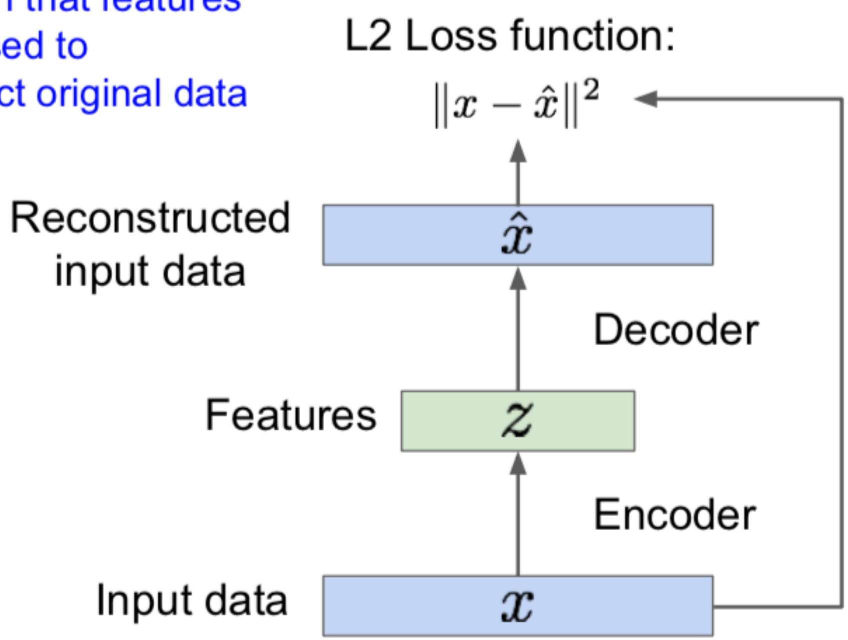


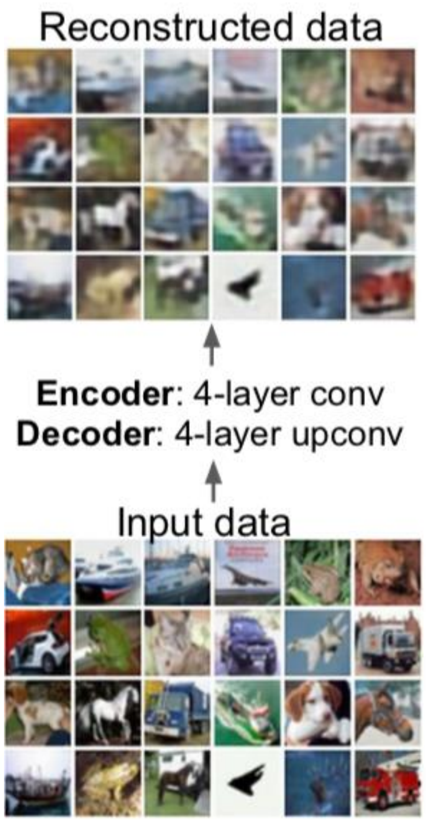


Autoencoders

- Training

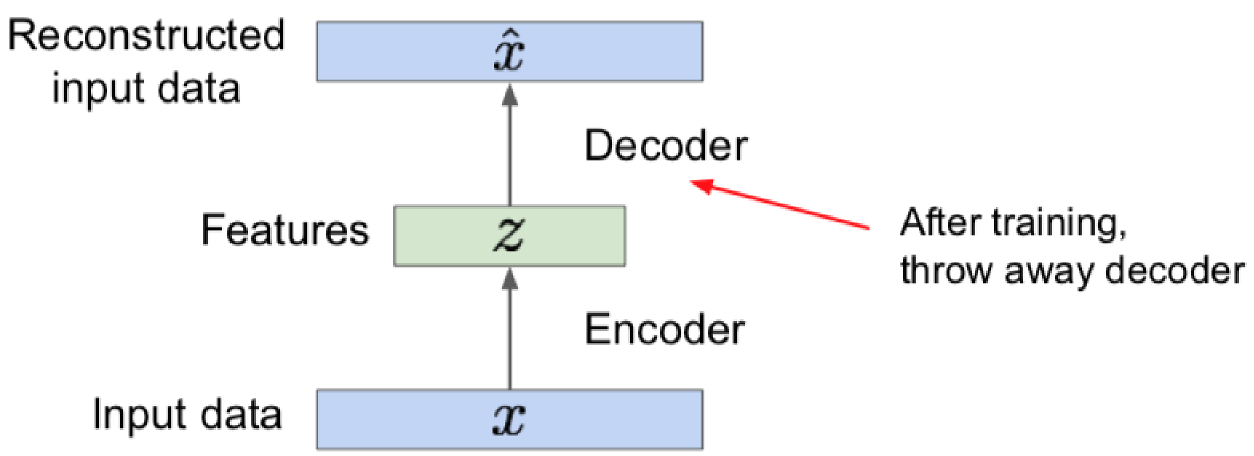
Train such that features
can be used to
reconstruct original data





Autoencoders

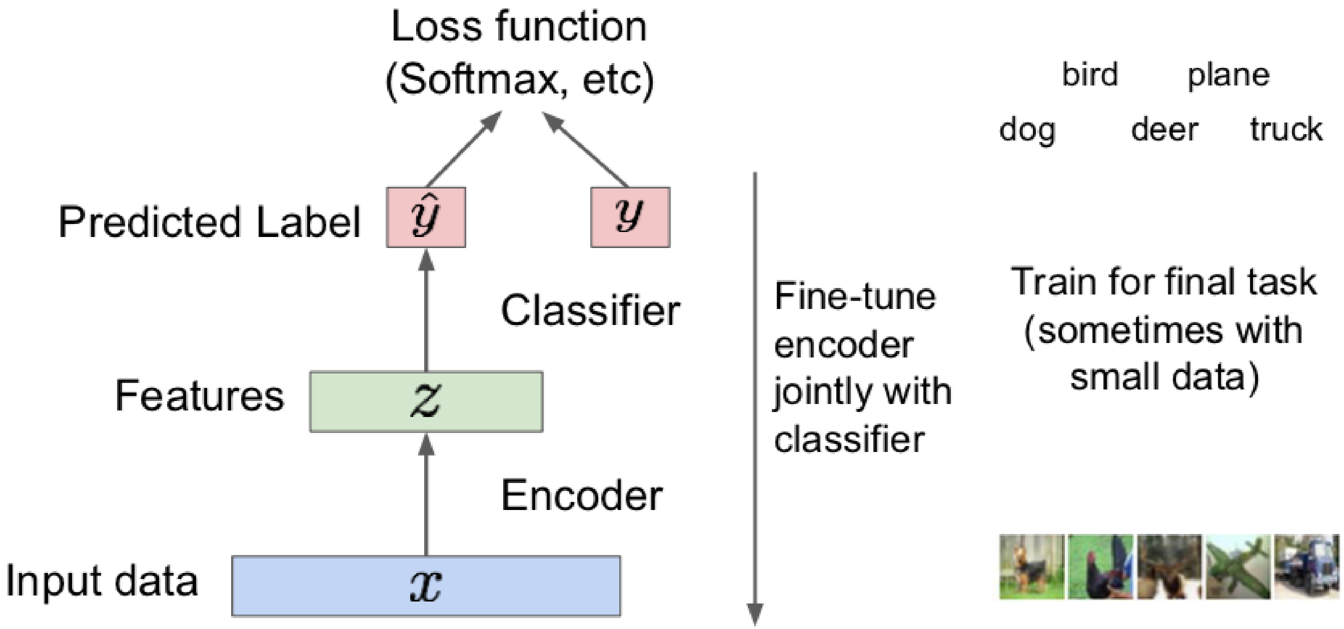
- Use the learned features for other tasks!



Autoencoders

- Use the learned features for other tasks!

Encoder can be used to initialize a **supervised** model



Avoid trivial solutions

- Undercomplete: dim(z) << dim(x)
  - Forces to capture the most salient features
  - Dimensionality reduction
  - Capture meaningful factors of variation
- Regularized
  - Encourage the model to have some properties

Sparse Autoencoders

- Code sparsity

$$LOSS = \|x - \hat{x}\|_2^2 + \|z\|_1$$

  - Helps learning good features for classification
  - Forces a (Laplace) prior on latent representation
  - Different from weight regularization! Why?

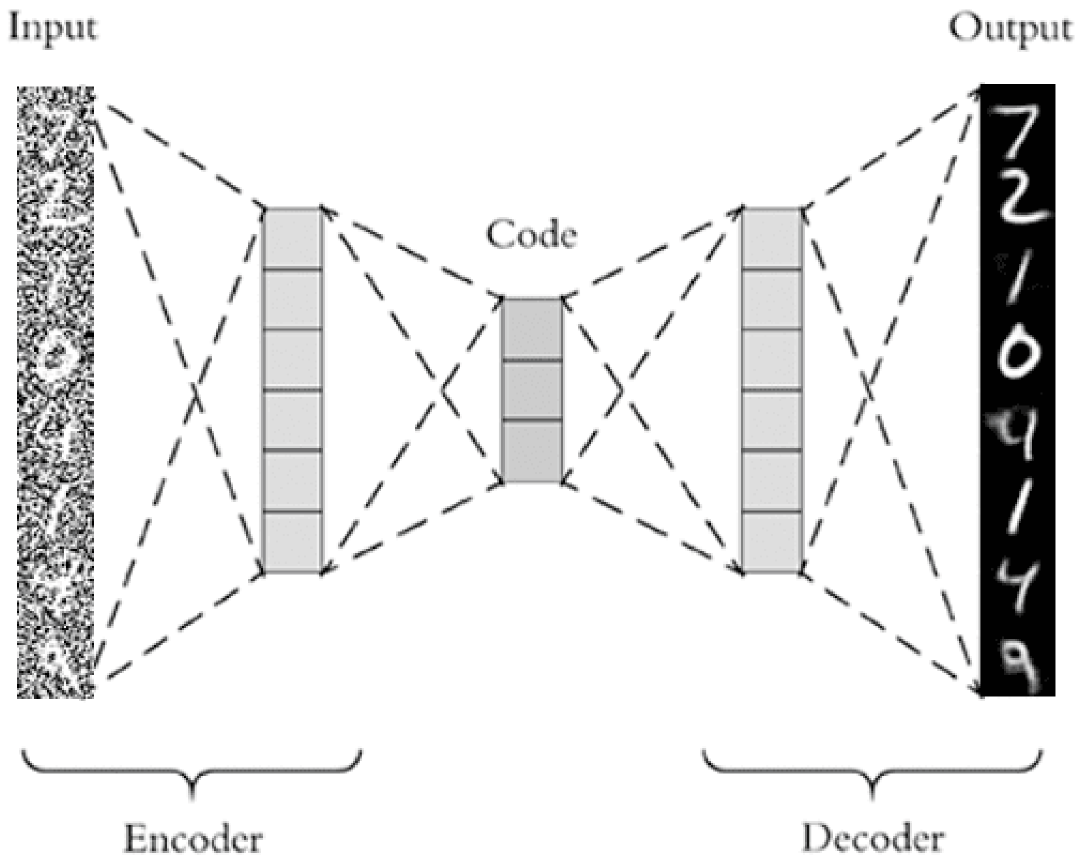
Denoising Autoencoders

- Definition
  - Encoder function: $z = E(x)$
  - Decoder function: $\hat{x} = D(z)$
  - Noisy version of data: $\tilde{x} = x + noise$
  - Denoising autoencoder:

$$LOSS_{den} = \|x - D(E(\tilde{x}))\|_2^2$$

- Implicitly learns the structure of the data

Denoising Autoencoders



https://www.pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/

Autoencoder Applications

- Dimensionality reduction
- Denoising
- Information retrieval
  - Low-dimensional, binary code (semantic hashing)
- Generative models
  - Variational autoencoders (VAEs)

Variational Autoencoders

- Idea: we can use the autoencoder approach to generate data from a specific distribution
- Training: data sampled from such distribution
- Use autoencoder to generate the statistical description of the data

Variational Autoencoders

- Generative model:
  – Given a set of training data, learn their distribution in order to generate new data from a similar distribution

Training data ~ $p_{data}(x)$ Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Variational Autoencoders

- Idea
  - Encoder and decoder provide **distributions** (their parameters), not data points!
- Assumptions
  - Training data $\{x_i\}_{i=1}^{N}$
  - $p(z)$ Gaussian distribution
  - $p(x|z)$ Gaussian distribution (Encoder)
  - $p(z|x)$ approximated by a Gaussian distribution (Decoder)

Variational Autoencoders

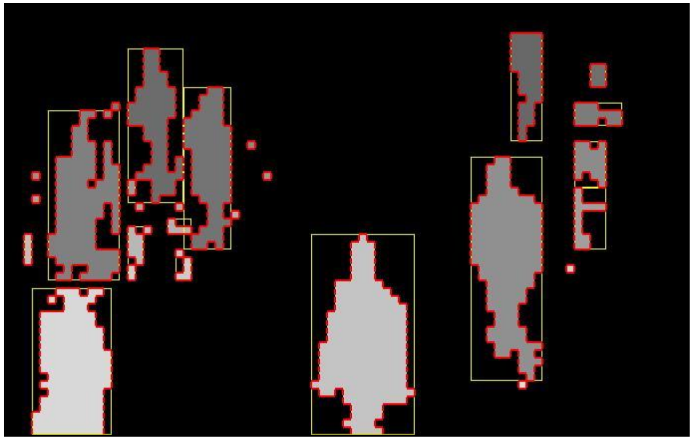
- Training
  - Use a variational lower bound of the log-likelihood $\log p(x_i)$
- Generate data
  - Sample z from a Gaussian prior
  - Use decoder to get (Gaussian) $p(x|z)$
  - Sample $x|z$ from $p(x|z)$

Outline

- Autoencoders
- Deep learning for segmentation

Semantic Segmentation

- Separation of the image in different areas
  - Objects
  - **Areas with similar** visual or **semantic characteristics**



First classify each pixel, and only then form regions (much harder!!)

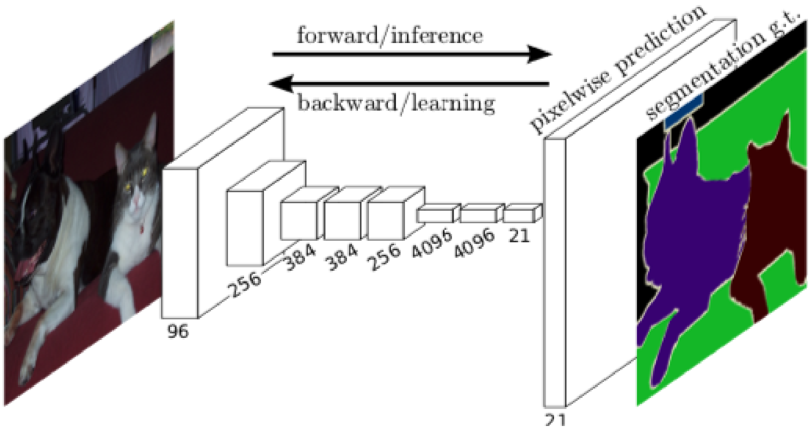
Deep Learning Semantic Segmentation

- Basic idea: use deep learning models to classify pixels with semantic labels
  - Can we simply use CNN architectures previously presented for classification?

- More demanding task than image classification

Fully Convolutional Networks

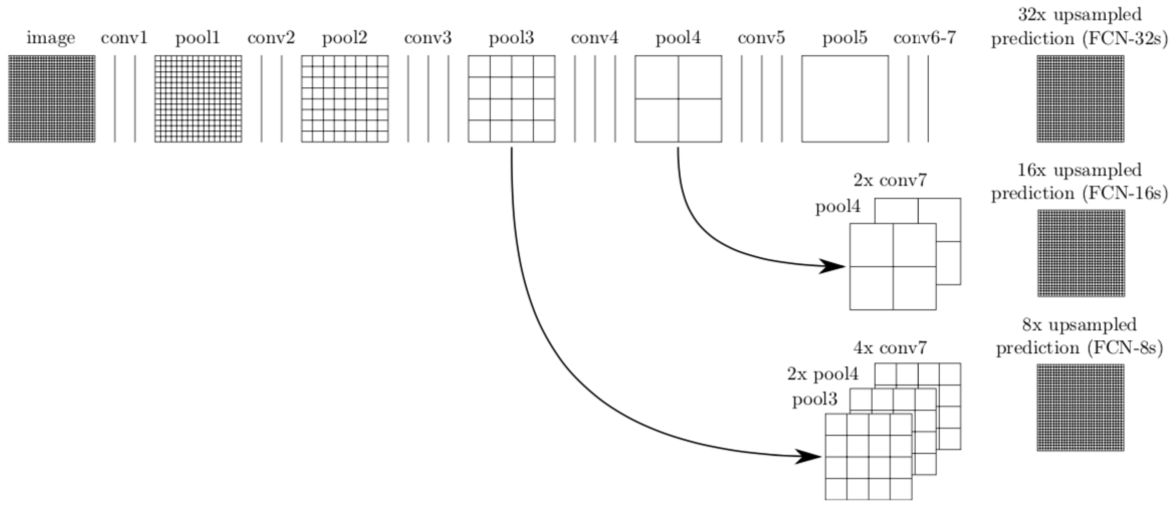
- Remove fully connected layers from existing CNN models (e.g., VGG16)
  - Variable size input
  - Output can have same size of input. (Why?)



J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

Fully Convolutional Networks

- Upsampling/Skip connections
  - Project information to image domain
  - Keep global information

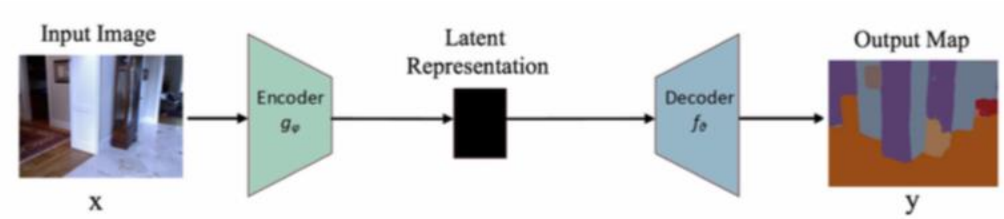


Fully Convolutional Networks

- Limitations:
  - Too complex for real time segmentation
  - Global information not efficiently managed
  - Not easily generalizable to 3D data

Encoder-Decoder Models

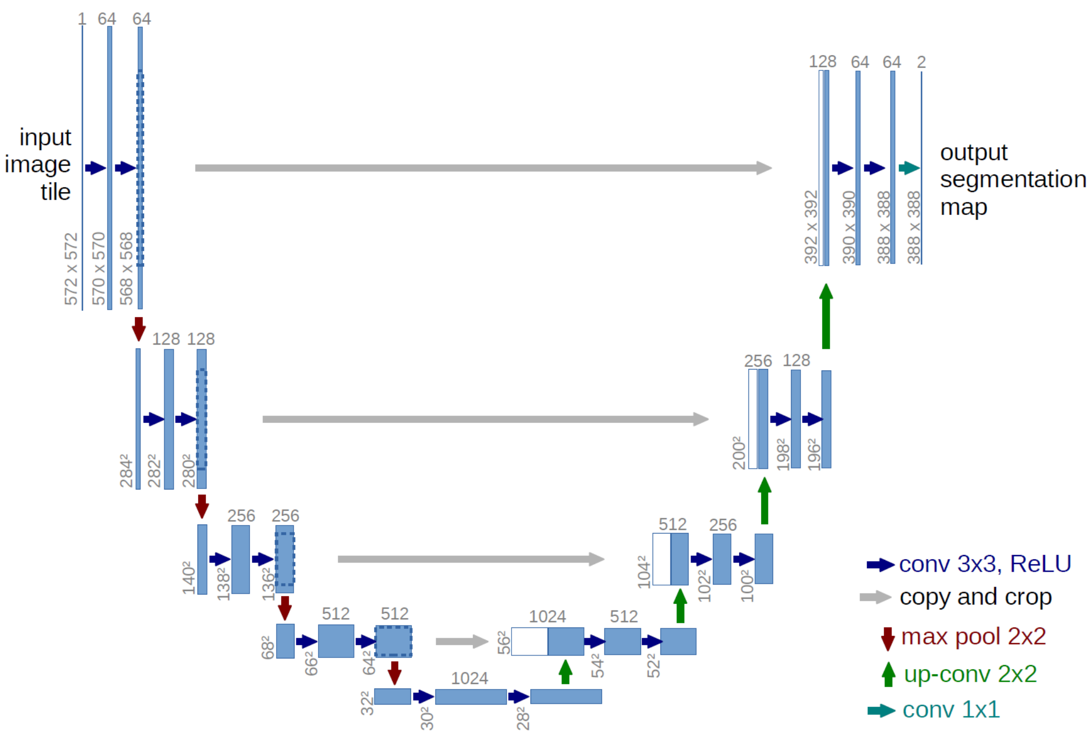
- Encoder-decoder architectures
  - Similar to autoencoders architectures
  - Leverage latent representation
  - But require labels to train (supervised)



Minaee S, Boykov YY, Porikli F, Plaza AJ, Kehtarnavaz N, Terzopoulos D. Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021 Feb 17

U-Net

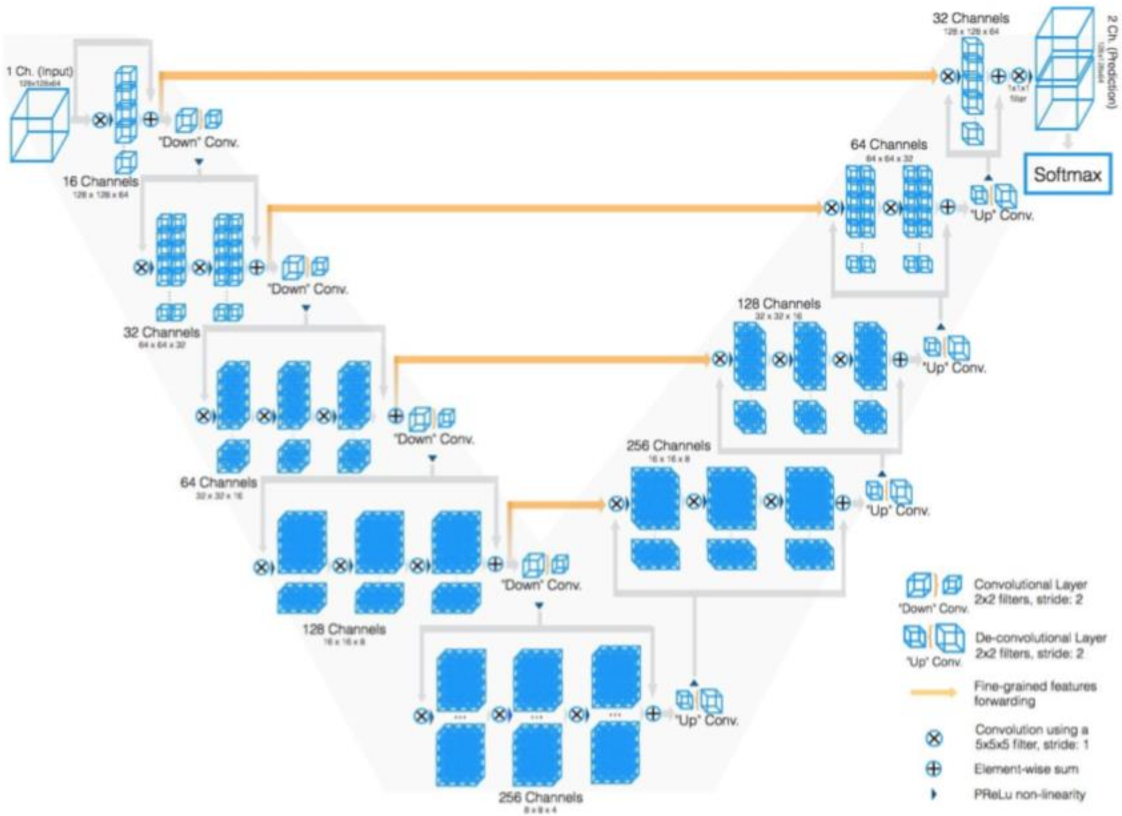
- 2D segmentation



O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation." In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241. Springer, Cham, 2015.

V-Net

- 3D segmentation



F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully convo- lutional neural networks for volumetric medical image segmen- tation,” in International Conference on 3D Vision. IEEE, 2016, pp. 565–571.

Encoder-Decoder Models

- Extensively used in as state-of-the-art for different fields
  - “General” image segmentation
  - Autonomous driving
  - Medical and biomedical image segmentation
- Limitations
  - Potential loss of fine-grained image information

Training

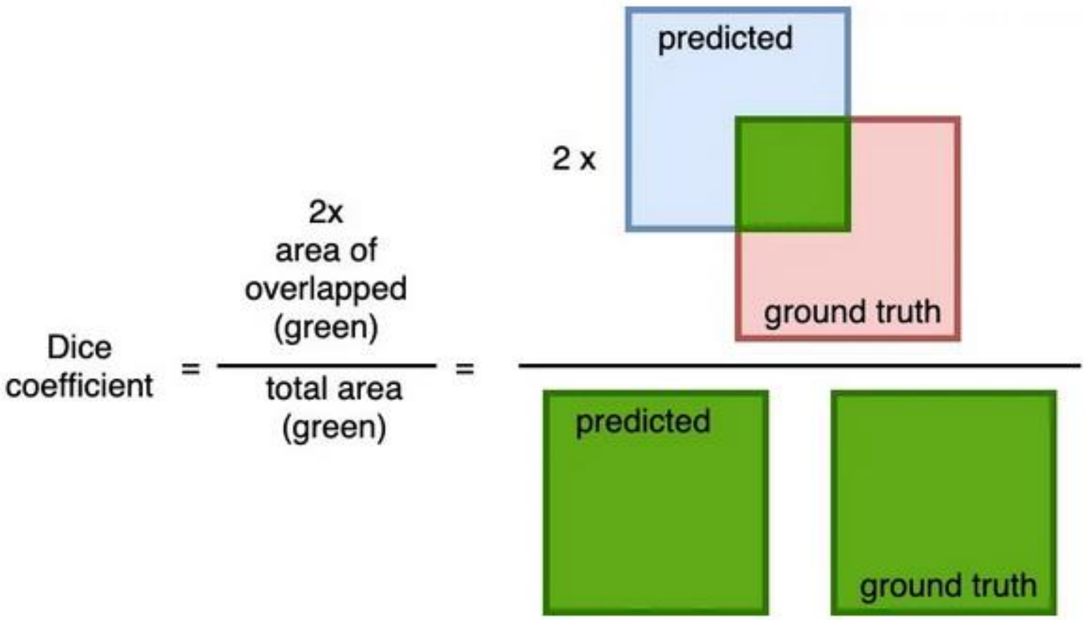
- Pixel classification
  - Pixel-level cross-entropy loss

$$CE_{loss} = -\frac{1}{N}\sum_{n=1}^{N} p_n \log q_n + (1-p_n)\log(1-q_n)$$

- Problem
  - Not very effective for highly imbalanced data

Training

- Dice coefficient



https://datascience.stackexchange.com/questions/75708/neural-network-probability-output-and-loss-function-example-dice-loss

Training

- Dice loss

$$DICE_{loss} = 1 - \frac{2\sum_{n=1}^{N} p_n q_n + \varepsilon}{\sum_{n=1}^{N} p_n + \sum_{n=1}^{N} q_n + \varepsilon}$$

- More robust against imbalanced data and directly related to “similarity” between the output segmentation map and true segmentation map

Resources

- F.F. Li, J. Johnson, S. Young. Convolutional Neural Networks for Visual Recognition, Stanford University, 2017
  - Lecture 13- ”Generative models”
  - http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
- I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Cambridge: MIT press, 2016.
  - Chapter 14 – “Autoencoders”
  - Chapter 20 – “Deep Generative Models”