Semantic segmentation

Diana Mateus

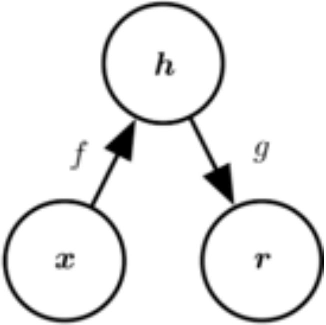
Table of contents

Autoencoders

Table of contents

Unsupervised learning with neural networks?

Autoencoders

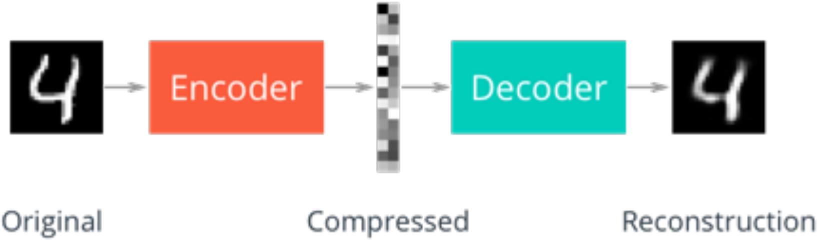


An **Autoencoder** is

- a Neural Network.
- with a **hidden layer** h describing a code.
- consist of two parts :
  - An encoder $h = f(x)$
  - A decoder $r = g(h)$
- seeks to reconstruct/copy the input.

$$g(f(x)) = x$$

Autoencoders - Motivation



Source: https://github.com/udacity/deep-learning/blob/master/autoencoder/Simple_Autoencoder_Solution.ipynb

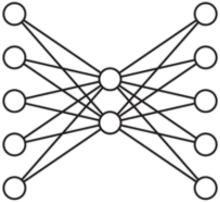
The code or compressed version of the image can be interpreted as its **latent representation**.

Autoencoders

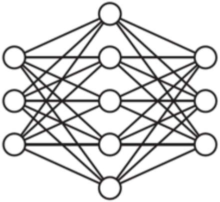
Architecture? Number of layers? Size of the code?

Autoencoders

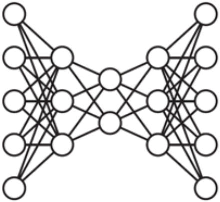
Architecture? Number of layers? Size of the code?



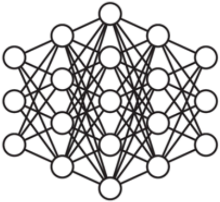
(a) Shallow undercomplete



(b) Shallow overcomplete



(c) Deep undercomplete

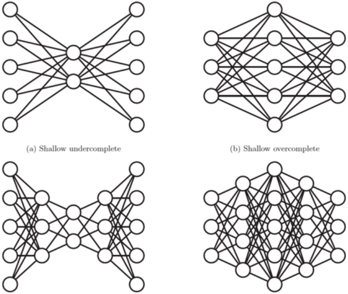


(d) Deep overcomplete

Source: David Charte, Francisco Charte, Salvador García, María J. del Jesus, Francisco Herrera, A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines, Information Fusion, Volume 44, 2018, Pages 78-96,

Autoencoders

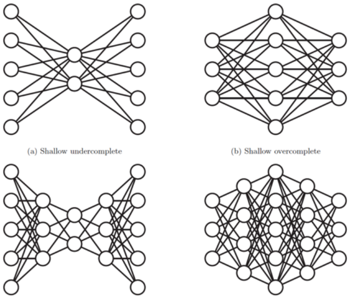
Loss?



(a) Shallow undercomplete (b) Shallow overcomplete

(c) Deep undercomplete (d) Deep overcomplete

Autoencoders


(a) Shallow undercomplete (b) Shallow overcomplete
(c) Deep undercomplete (d) Deep overcomplete

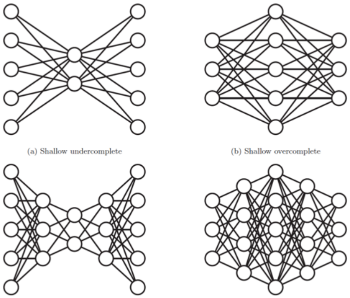
Loss?

- **Undercomplete:** reduce the dimensionality, capture salient features from data.

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

e.g. MSE
If too much capacity it may learn a look up table instead of a meaningful representation.

Autoencoders



(a) Shallow undercomplete (b) Shallow overcomplete (c) Deep undercomplete (d) Deep overcomplete

Loss?

- **Undercomplete:** reduce the dimensionality, capture salient features from data.

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

e.g. MSE

- **Overcomplete** encourage the model to have other properties beyond copying.

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

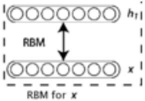
e.g. MSE + sparsity constraint.

Table of contents

Autoencoders – modern history

Restricted Boltzmann Machines (RBM) are remotely linked to autoencoders but share some similar training mechanics. Different to AEs, RBMs rely on **stochastic** units. RBMs are useful for **unsupervised** learning of the **data distribution.**

Input: $\{\mathbf{x}_i\}_{i=1}^N$, Output: $\{\mathbf{h}_i\}_{i=1}^N$



Source: http://www.cs.toronto.edu/~rsalakhu/deeplearning/yoshua_icml2009.pdf

$$p(h_j = 1|\mathbf{x}) = \sigma(c_j + \sum_{i=1}^m w_{ij} \cdot x_j)$$
$$p(x_i = 1|\mathbf{h}) = \sigma(b_j + \sum_{j=1}^n w_{ij} \cdot h_j)$$

Goal: Learn undirected weights w_{ij} (Log Lik. + gradient descent).

Learning: repeat and adjust the weights to minimize error.

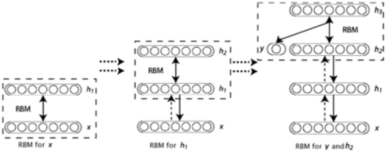
2 Alternating Gibbs samplings steps:

- *propagate*: sample hiddens $\mathbf{h}$ given visibles $\mathbf{x}$;
- *reconstruct*: sample visibles $\mathbf{x}$ given hiddens $\mathbf{h}$;

Autoencoders – modern history

Restricted Boltzmann Machines (RBM) are remotely linked to autoencoders but share some similar training mechanics. Different to AEs, RBMs rely on **stochastic** units. RBMs are useful for **unsupervised** learning of the **data distribution.**

Stacking (RBMs) = **Deep Belief Network** (DBN):

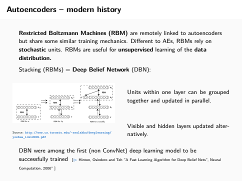


Source: http://www.cs.toronto.edu/~rsalakhu/deeplearning/yoshua_icml2009.pdf

Units within one layer can be grouped together and updated in parallel.

Visible and hidden layers updated alternatively.

DBN were among the first (non ConvNet) deep learning model to be successfully trained [▷ Hinton, Osindero and Teh "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 2006"]



Boltzmann machines were introduced during the connectionist wave (1983-1986) as an approach to learning arbitrary **probability distributions** over **binary** vectors $\mathbf{x} \in \{0, 1\}^d$

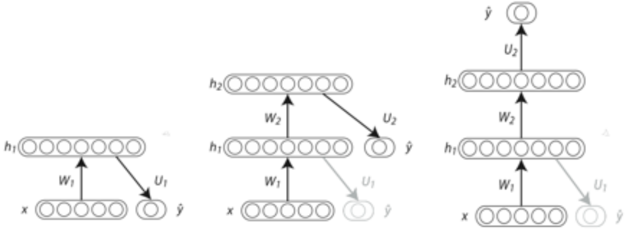
Restricted Boltzman machines (RBM) (Smolensky, 1986) are undirected **probabilistic** graphical models containing a layer of observable variables and a single layer of latent variables. They rely on **stochastic** (binary) units with a given (usually binary of Gaussian) distribution.

They are **restricted** in the sense that no connection is allowed between inputs or neurons of the same layer.

RBMs can be stacked to form a **deep belief network (DBN)** or with some modifications a Deep Boltzman Machine (DBM).

Autoencoders – Modern history

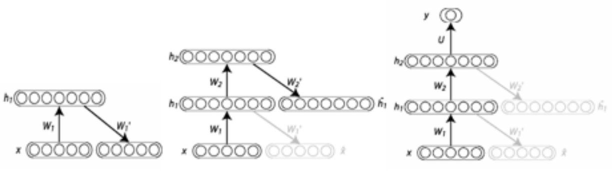
Greedy layer-wise supervised pre-training



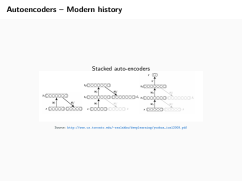
Source: http://www.cs.toronto.edu/~rsalakhu/deeplearning/yoshua_icml2009.pdf

Autoencoders – Modern history

Stacked auto-encoders



Source: http://www.cs.toronto.edu/~rsalakhu/deeplearning/yoshua_icml2009.pdf



Similar to DBNs, **Stacked Autoencoders** (SAEs) are built by stacking single-level autoencoders to create a deep network.

Up to 2006 deep networks were thought to be too difficult to train and as such of limited utility. The breakthrough came with

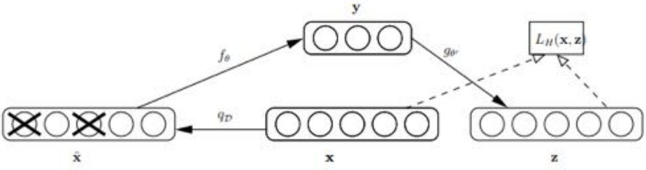
- [▷ Hinton, Osindero and Teh "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 2006"]
- [▷ Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », NIPS'2006]
- [▷ Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », NIPS'2006]

These references showed that fast, layerwise greedy and unsupervised algorithms could be used to initialize the weights of a deep network. The slower algorithm needed only to **fine tune** the learned weights to provide good results.

Table of contents

Autoencoders – Recent Advances

Stacked Denoising autoencoders



[▷ Vincent, Pascal, et al. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion." Journal of Machine Learning Research 11 (2010): 3371-3408.]

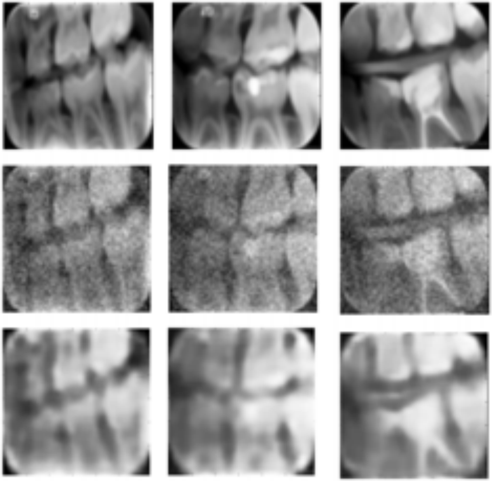
Autoencoders – Examples with CNNs

Stacked Denoising autoencoders

Source: http://www.opendeep.org/v0.0.5/docs/tutorial-your-first-model

Autoencoders – Examples with CNNs

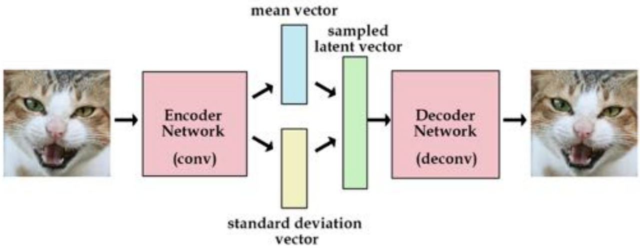
Stacked Denoising autoencoders



Source: https://arxiv.org/pdf/1608.04667.pdf

Autoencoders – Recent Advances

Variational autoencoders



http://kvfrans.com/variational-autoencoders-explained/

[▷ Doersch C. Tutorial on Variational Autoencoders;. Available from: https://arxiv.org/abs/1606.05908.]

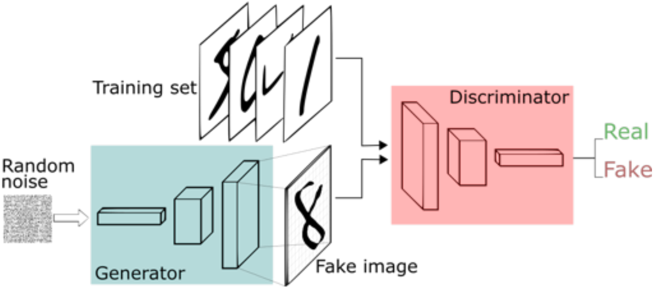
Autoencoders – Examples with images

Variational autoencoders

https://github.com/WojciechMormul/vae

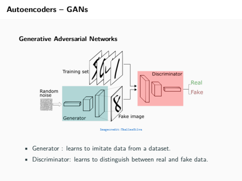
Autoencoders – GANs

Generative Adversarial Networks



Imagecredit:ThallesSilva

- Generator : learns to imitate data from a dataset.
- Discriminator: learns to distinguish between real and fake data.



Two player game

$$\min \max(D, G)$$

The **discriminator** D is a classifier trained directly on real and generated images and is responsible for classifying images as real or fake (generated).

$$\max_{W_D} \log D(\mathbf{x}) + \log(1 - D(G(\mathbf{z})))$$

The **generator** G is not trained directly and instead is trained via the discriminator model.

$$min_{W_G} \log(1 - D(G(\mathbf{z})))$$

The discriminator is learned to provide the loss function for the generator.

Equilibrium between generator and discriminator loss is sought.

Table of contents

We can also condition the sampled predictions to another input. For instance in BlendGAN [Liu et.al NIPS 2021]https://github.com/onion-liu/BlendGAN the output is conditioned to an image-style pair:

Input - Style - Output

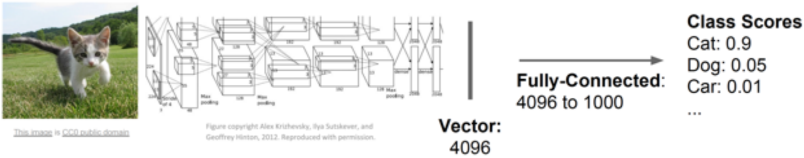
Table of contents

Image Segmentation

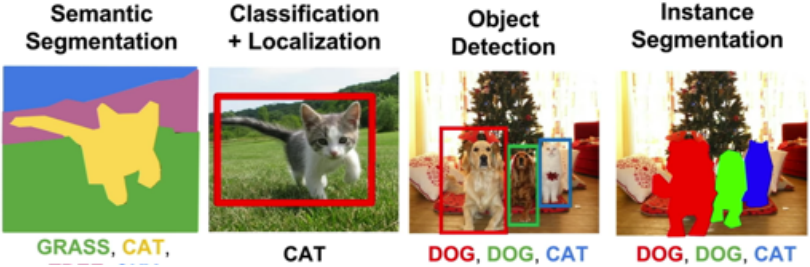
Table of contents

Table of contents

Image Classification



Computer vision tasks



Source: Standford cs231, lecture 11: Detection and segmentation

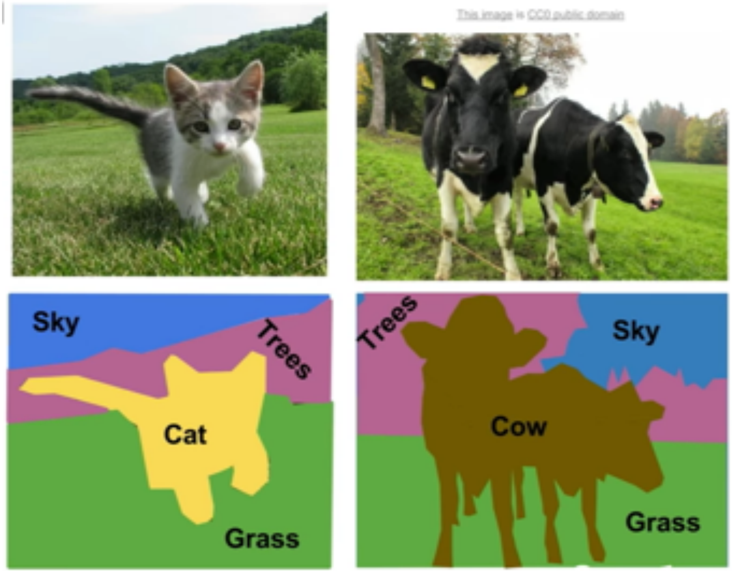
Definition of Semantic Segmentation

Source:

https://sthalles.github.io/deep_segmentation_network/

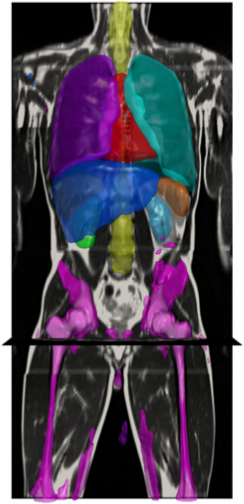
- **Input**: image
- **Output**: decide the category of each pixel (not each image).

Semantic Segmentation vs. Instance segmentation



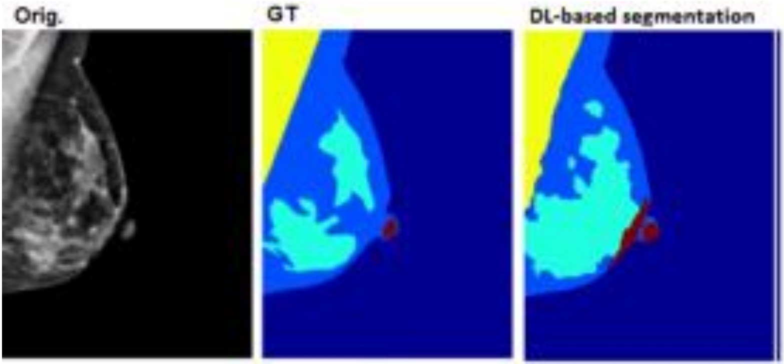
Source: Standford cs231, lecture 11: Detection and segmentation

Semantic Segmentation for medical image analysis



Source: https://wiki.tum.de/display/lfdv/Image+Semantic+Segmentation

Semantic Segmentation for medical image analysis



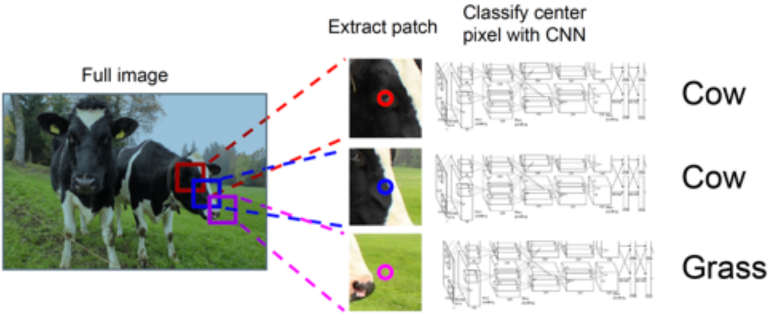
Source: http://www.research.ibm.com/haifa/dept/imt/mia_research.shtml

Table of contents

How do we do Semantic Segmentation with a CNN?

How do we do Semantic Segmentation with a CNN?

Patch-wise segmentation.



Source: Standford cs231, lecture 11: Detection and segmentation

[▷ Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013 Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014]

How do we do Semantic Segmentation with a CNN?

Patch-wise segmentation.

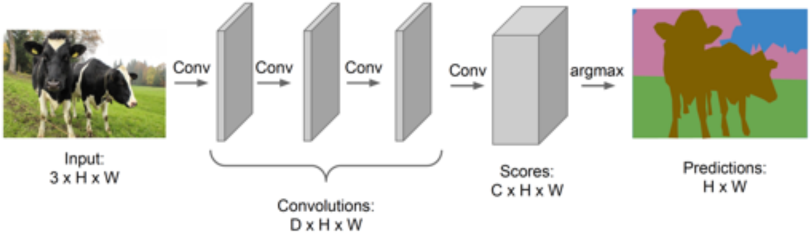
Problems:

- Inefficient.
- Independent computations for neighbouring pixels.
- Not reusing shared features between overlapping patches.

Solution?

Semantic Segmentation – Fully Convolutional Networks

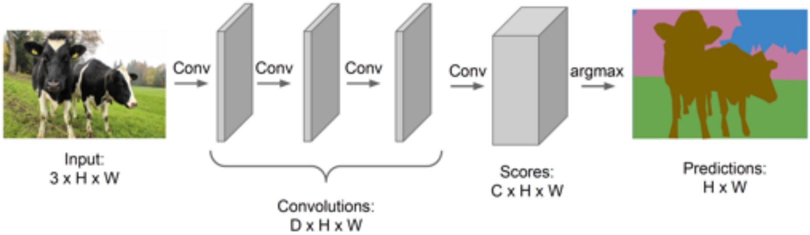
Design a network as a bunch of convolutional layers to make predictions for pixels all at once



Source: Standford cs231, lecture 11: Detection and segmentation

Semantic Segmentation – Fully Convolutional Networks

Design a network as a bunch of convolutional layers to make predictions for pixels all at once



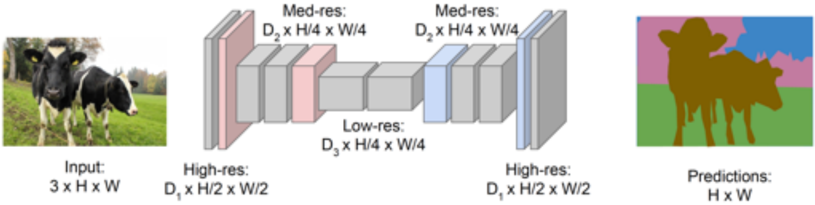
Source: Standford cs231, lecture 11: Detection and segmentation

However:

- Very expensive due to resolution preservation.
- Memory usage is very high.

Ideas?

Semantic Segmentation – Fully Convolutional Networks

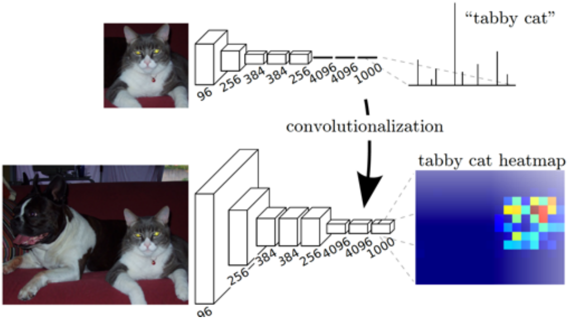


Source: Standford cs231, lecture 11: Detection and segmentation

[▷ Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.]

- Goal: predictions of the **same size** as the original input.
- Idea: design a fully convolutional network with downsampling and upsampling inside.

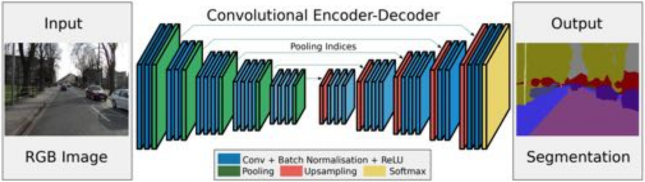
Semantic Segmentation – Fully Convolutional Networks



[▷ Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.]

- Replace fully connected layers with convolutional layers.
- Final convolutional layer is a tensor of size $C \times H \times W$, where C is the number of categories.
- Result can be interpreted as heatmap.

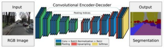
Semantic Segmentation – Fully Convolutional Layer



[▷ Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv preprint arXiv:1511.00561 (2015).]

- Use an encoder-decoder structure but using convolutional layers.
- Deeper networks possible yet less computations needed due to reduced resolution.

Semantic Segmentation – Fully Convolutional Layer



[▷ Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv preprint arXiv:1511.00561 (2015).]

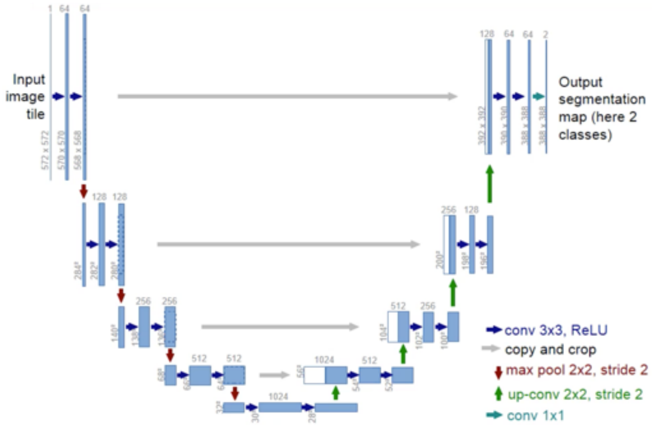
- Use an encoder-decoder structure but using convolutional layers.
- Deeper networks possible yet less computations needed due to reduced resolution.

Decrease the spatial resolution of the predictions and then increase it in the second half so that the output can have the same size as the input.

Faster than patch-wise segmentation. An additional advantage is that a pre-trained CNN for classification can be used for the encoder portion of the network.

Also [▷ Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015]

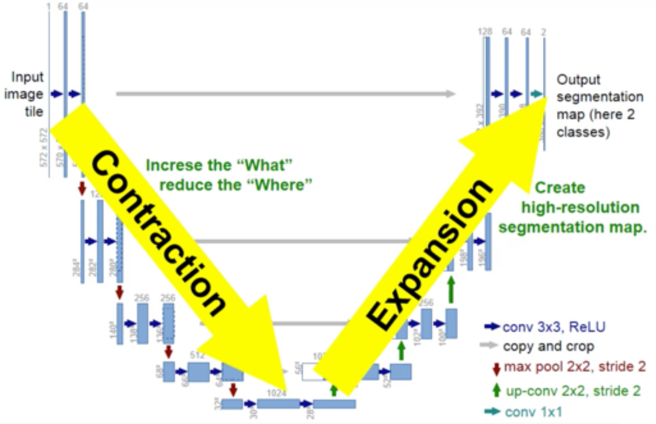
Semantic Segmentation – U-net



[▷ Ronneberger, Olaf, Philipp Fischer, and Thomas Brox."U-net: Convolutional networks for biomedical image segmentation" Int. Conf. on Medical image computing and computer-assisted intervention. Springer 2015.]

Introduce **skip connections** to increase the precision at borders lost during the contraction.

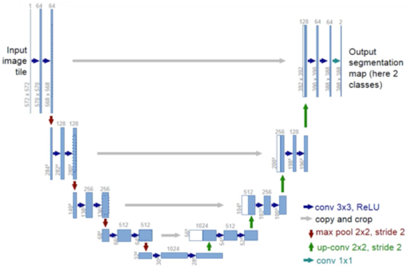
Semantic Segmentation – U-net



[▷ Ronneberger, Olaf, Philipp Fischer, and Thomas Brox."U-net: Convolutional networks for biomedical image segmentation" Int. Conf. on Medical image computing and computer-assisted intervention. Springer 2015.]

Introduce **skip connections** to increase the precision at borders lost during the contraction.

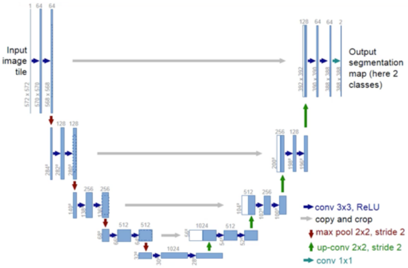
Semantic Segmentation – U-net



Challenges

- 30 annotated images
- Touching objects

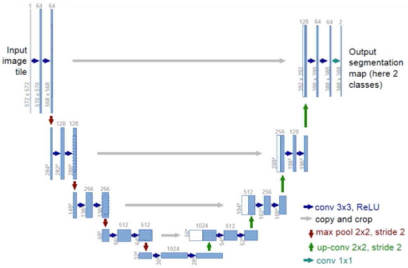
Semantic Segmentation – U-net



Architecture

- End-to-end
- Only valid convolutions
- ReLU
- Max pooling

Semantic Segmentation – U-net



Training and testing

- Special loss for borders
- 10h/training
- 1s/image test

Table of contents

Semantic Segmentation – Components

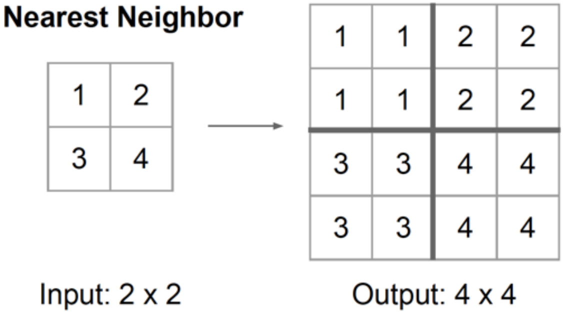
Encoder: for decreasing resolution.

- Pooling(avg, max).
- Strided convolutions.

How to invert

Semantic Segmentation – Components

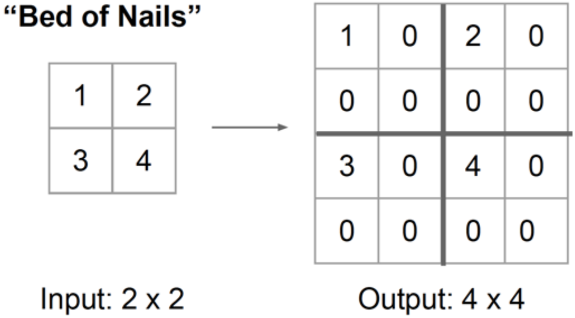
Unpooling



Source: Standford cs231, lecture 11: Detection and segmentation

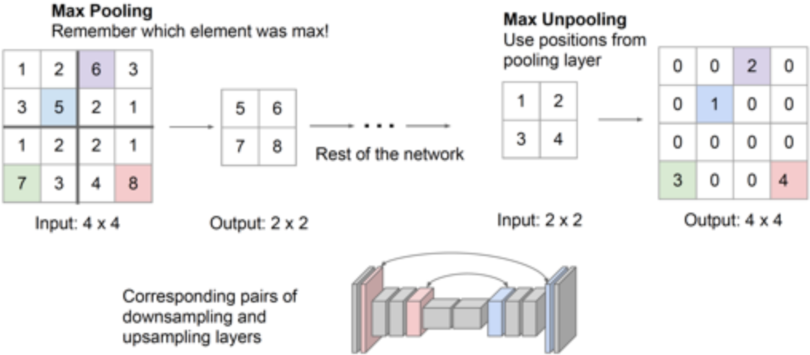
Semantic Segmentation – Components

Unpooling



Source: Standford cs231, lecture 11: Detection and segmentation

Semantic Segmentation – Unpooling



Source: Standford cs231, lecture 11: Detection and segmentation

Semantic Segmentation – Components

Transpose Convolution

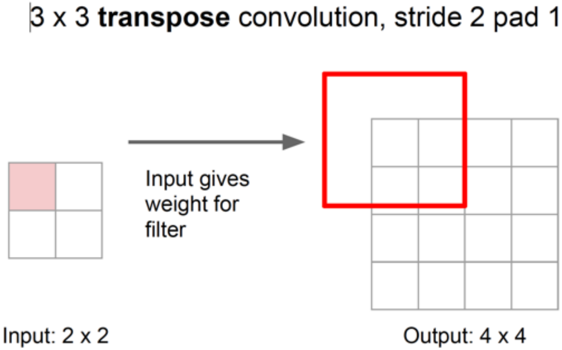
3 x 3 **transpose** convolution, stride 2 pad 1

Input: 2 x 2 Output: 4 x 4

Source: Standford lecture cs231

Semantic Segmentation – Components

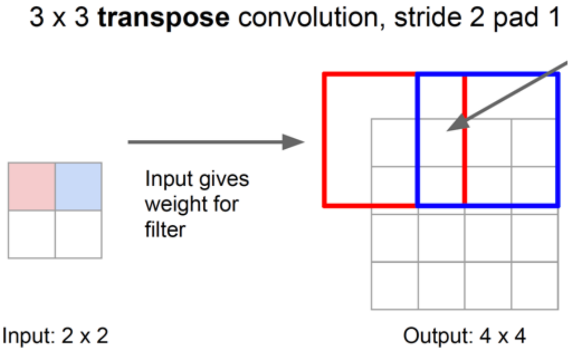
Transpose Convolution



Source: Standford lecture cs231

Semantic Segmentation – Components

Transpose Convolution



Source: Standford lecture cs231

Semantic Segmentation – Components

Transpose Convolution

Convolution and Transpose Convolution : 1D example

Example: 1D conv, kernel size=3, stride=2, padding=1

Convolution

$\mathbf{w} * \mathbf{x} = \mathbf{W}\mathbf{x}$

$$\mathbf{w} * \mathbf{x} = \begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \end{bmatrix}$$

$$\mathbf{w} * \mathbf{x} = \begin{bmatrix} w_2 x_1 + w_3 x_2 \\ w_1 x_1 + w_2 x_2 + w_3 x_3 \\ w_1 x_2 + w_2 x_3 + w_3 x_4 \\ w_1 x_3 + w_2 x_4 \end{bmatrix}$$

Transpose Convolution

$\mathbf{w} *^{\top} \mathbf{z} = \mathbf{W}^{\top}\mathbf{z}$

$$\mathbf{w} *^{\top} \mathbf{z} = \begin{bmatrix} w_1 & 0 & 0 & 0 \\ w_2 & w_1 & 0 & 0 \\ w_3 & w_2 & w_1 & 0 \\ 0 & w_3 & w_2 & w_1 \\ 0 & 0 & w_3 & w_2 \\ 0 & 0 & 0 & w_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

$$\mathbf{w} *^{\top} \mathbf{z} = \begin{bmatrix} w_1 z_1 \\ w_2 z_1 + w_1 z_2 \\ w_3 z_1 + w_2 z_2 + w_1 z_3 \\ w_3 z_2 + w_2 z_3 + w_1 z_4 \\ w_3 z_3 + w_2 z_4 \\ w_3 z_4 \end{bmatrix}$$

Transpose Convolution 2D

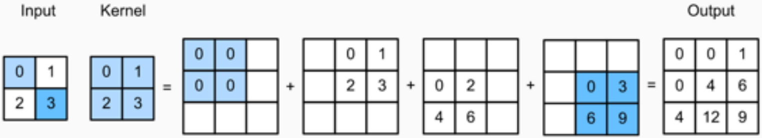
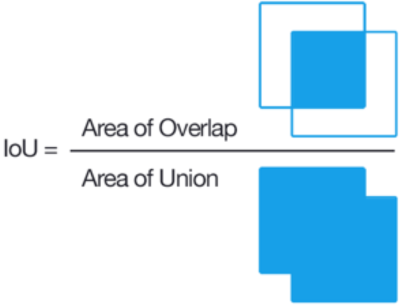


Fig. 12.10.1 Transposed convolution layer with a 2×2 kernel.

Table of contents

Segmentation loss and evaluation measures

Intersection over Union

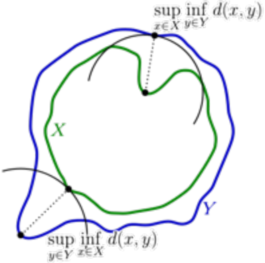


Segmentation loss and evaluation measures

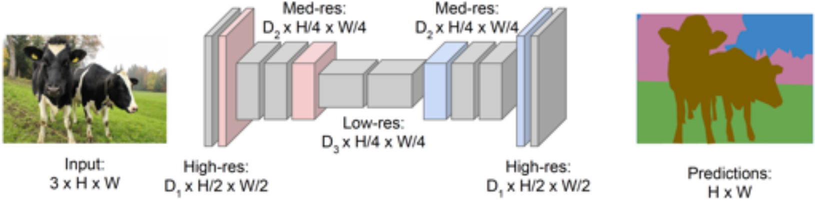
- Dice score

$$\frac{P \cap T}{(|P| + |T|)/2}$$

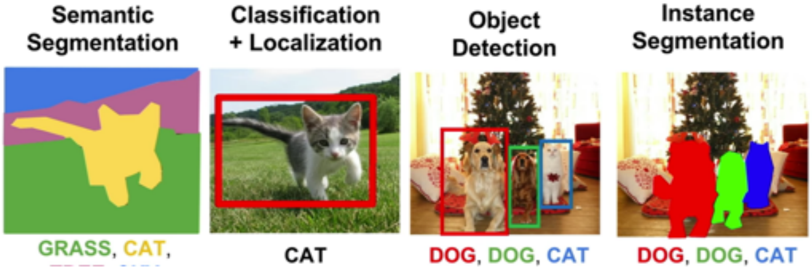
- Hausdorff distance



Semantic Segmentation – Summary FCN



Computer vision tasks



Summary

References

- “Deep Learning”. Book by Aaron Courville, Ian Goodfellow, and Yoshua Bengio.
- History of CNNs https://arxiv.org/pdf/1803.01164.pdf
- Network types https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f236746
- Hands-On Machine Learning with Scikit-Learn