APS360 Fundamentals of AI

Lisa Zhang

Lecture 9; Feb 4, 2019

Agenda

- Fully Convolutional Architectures
- Deconvolutions
- Autoencoders

Autoencoder

Fully-convolutional architectures

- Architectures without fully-connected layers
- Useful for generalizing to images of any size
- Also useful for pixel-wise predictions

Pixel-wise prediction

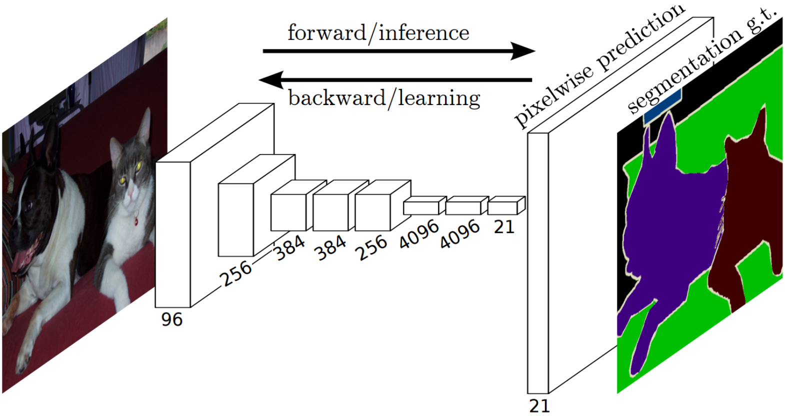


Figure 1: http://deeplearning.net/tutorial/fcn_2D_segm.html

Question:

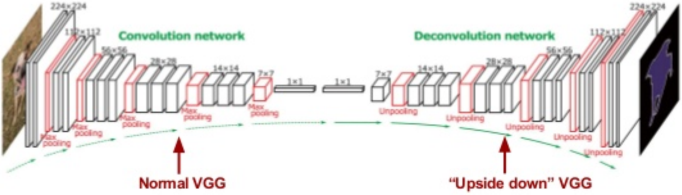
- How to generate pixel-wise prediction?
- How to generate images?

Transpose Convolution

More than one upsampling layer

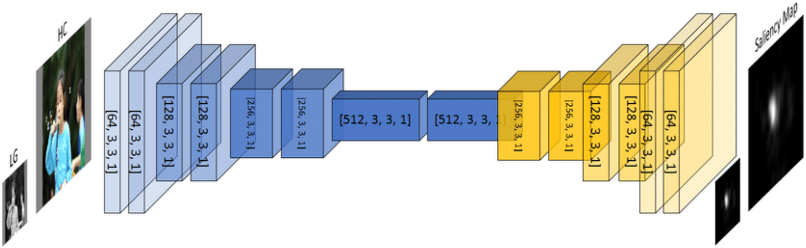
DeconvNet:

VGG-16 (conv+Relu+MaxPool) + mirrored VGG (Unpooling+'deconv'+Relu)

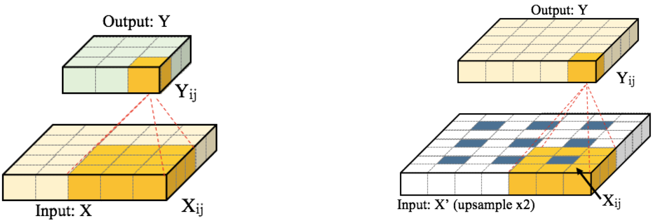


Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Transpose Convolution



Transpose Convolution Layer



(**a**) Convolutional layer: the input size is $W_1 = H_1 = 5$; the receptive field $F = 3$; the convolution is performed with stride $S = 1$ and no padding ($P = 0$). The output Yis of size $W_2 = H_2 = 3$.

(**b**) Transposed convolutional layer: input size $W_1 = H_1 = 3$; transposed convolution with stride $S = 2$; padding with $P = 1$; and a receptive field of $F = 3$. The output Yis of size $W_2 = H_2 = 5$.

Figure 2: https://www.mdpi.com/2072-4292/9/6/522/htm

Better Visual

https://github.com/vdumoulin/conv_arithmetic

Transpose Convolutions in PyTorch

```
nn.ConvTranspose2d(in_channels=16,
                   out_channels=8,
                   kernel_size=5,
                   stride=2,
                   output_padding=1, # needed because stric
                   padding=2)
```

Example: Autoencoder

To demonstrate ConvTranspose2d, we will build a network that:

- Finds a lower dimensional representation of the image
- Then reconstructs the image from the low-dimensional representation

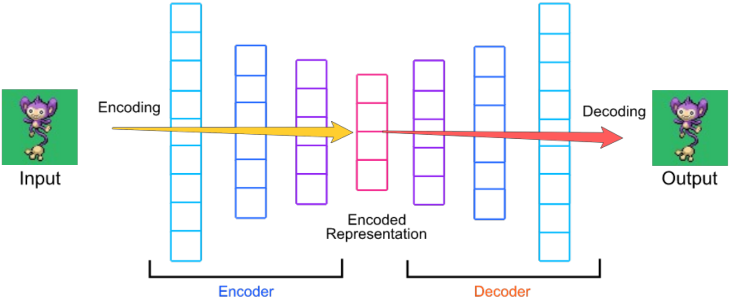
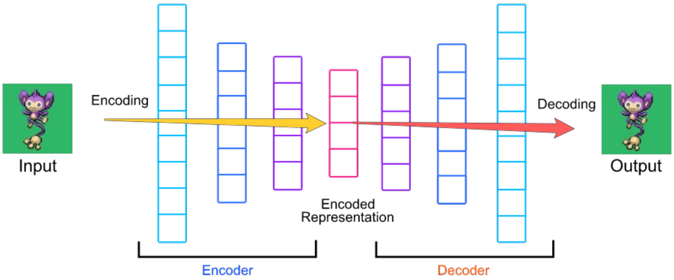


Figure 3: https://hackernoon.com/how-to-autoencode-your-pok%C3%A9mon-6b0f5c7b7d97

The components of an autoencoder



Encoder:

- Input = image
- Output = low-dimensional embedding

Decoder:

- Input = low-dimensional embedding
- Output = image

Why autoencoders?

- Dimension reduction:
  - find a low dimensional representation of the image
- Image Generation:
  - generate new images not in the training set

How to train autoencoders?

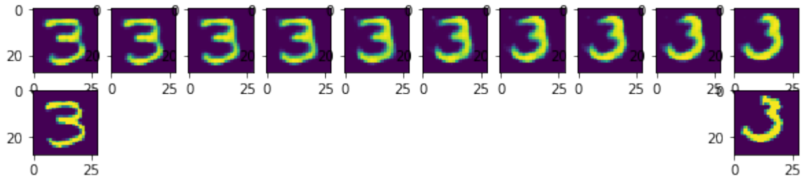
- Loss function:
  - How close were the reconstructed image from the original?
  - **Mean Sqaure Error Loss**: look at the mean square error across all the pixels.
- Optimizer:
  - Just like before!
  - Introduce a new optimizer: `Adam`
  - Commonly used for other network architectures too
- Training loop:
  - Just like before!

Structure in the Embedding Space

The dimensionality reduction means that there will be structure in the embedding space.

If the dimensionality of the embedding space is not too large, similar images should map to similar locations.

Interpolating in the Embedding Space



Recommended Reading: Issues with Transpose Convolutions

https://distill.pub/2016/deconv-checkerboard/