CS281: Advanced ML **November 20, 2017**

Lecture 21: Deep Learning in Health Care

Lecturer: Sasha Rush *Scribes: Raghu Dhara, Rui Fang*

We had a guest lecturer today: Sumit Chopra, head of AI Research at Imagen Technologies, a stealth-phase medical imaging startup.

21.1 Past Work

Before his work at Imagen, Dr. Chopra was a research scientist at Facebook AI Research. There, Dr. Chopra worked primarily in natural language processing (with Dr. Rush!). He completed his doctoral thesis under Yann LeCun at NYU. Here, he worked on Siamese Networks for picture distance metrics, DrLIM (dimensionality reduction by learning an invariant mapping) , and factor graphs for relational regression, a topic he then later applied at a startup to predict residential real estate prices. He has also worked at AT&T labs as a research scientist.

21.2 Computer Vision

21.2.1 Traditional Methods

Traditional methods in computer vision perform image classification by transforming an input image into features through a hand-crafted feature extractor (SIFT, HOG, ...) and then classifying the features using a classifier (SVMs, Neural Network (rarely), ...). The drawback of such methods is that hand crating features become challenging when input is not visually perceptible: depth map, etc.

21.2.2 Deep Learning: Convolutional Neural Networks (CNNs)

Unlike traditional methods, deep learning proposes to learn the features from scratch - learn everything end to end. The intuition behind deep learning is to learn a highly complex function composed of lots of simple functions, of which the parameters will be learned. The main class of deep learning architectures used in computer vision is the convolutional neural networks.
There are three basic types of hidden layers in a CNN: convolutional layer, non-linearity layer, and pooling layer.

- The convolutional layer is built upon convolution operation, which applies a kernel matrix to an input matrix and returns the dot products between the kernel and each receptive field. For example, Figure 21.1 shows applying a 3×3 kernal with stride $= 1$ (move one pixel at a time) over a 7×7 input matrix results in a 5×5 output matrix.
- The non-linearity layer applies a non-linear activation function to the output of the convolutional layer.
- The pooling layer works as subsampling: it applies a kernel (usually a max function) to an input matrix without overlapping, as shown in Figure 21.2. The pooling layer reduces spatial dimension of the input and results in spatial invariance.

A typical CNN architecture is shown in Figure 21.3. This CNN performs an image level classification - it takes the entire image and output predictions of the major theme of this image. The input image is convoluted and subsampled a few times before getting classified through the fully connected layers.

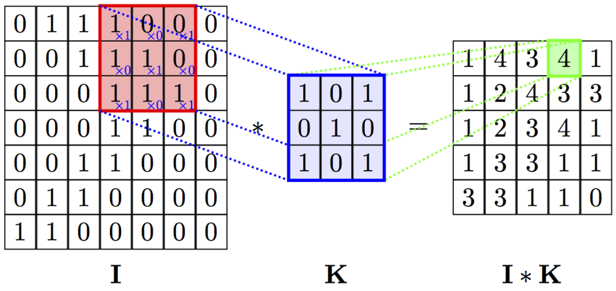


Figure 21.1: Convolution

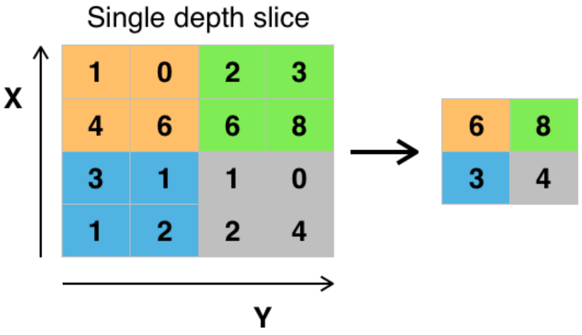


Figure 21.2: Pooling

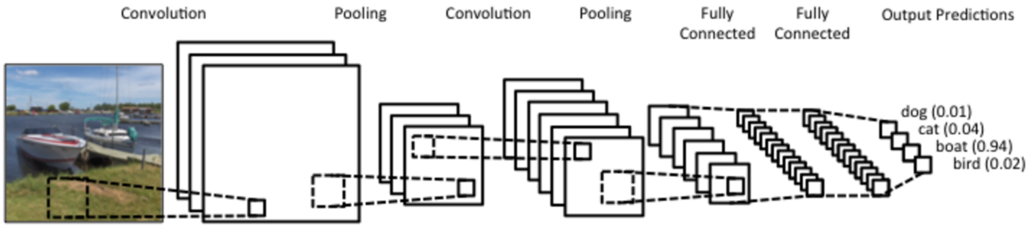


Figure 21.3: Typical CNN architecture

21.2.3 Pixel Level Classification CNNs

For image recognition, the tasks range from image level classification to object level classification to pixel level classification. The focus here is pixel level classification: the convolutional neural networks have a loss function associated with every pixel. Applications of pixel level classification include scene understanding, semantic segmentation, depth map prediction, medical imaging, etc.
We introduce three CNN based architectures in this category:

1. Fully Convolutional Network (FCN) [1]

   Fully convolutional networks take input of arbitrary size and produce correspondingly sized output (see Figure 21.4). Different from typical classification CNN architectures where outputs are non-spatial, in FCN the classification layers (fully-connected layers) are viewed as convolutions with kernels that cover their entire input regions, therefore generating 2D classification maps as outputs (see Figure 21.5). While the output is 2D, it is still coarse due to subsampling. Hence, upsampling is needed to generate outputs of the same size as inputs. In addition, the networks are designed to combine predictions from both the final layer and the intermediate layer to provide finer details (see Figure 21.6 and Figure 21.7).

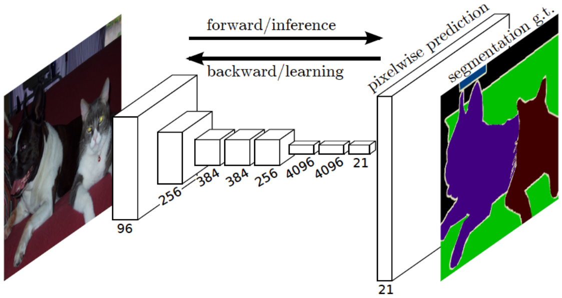


Figure 21.4: Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

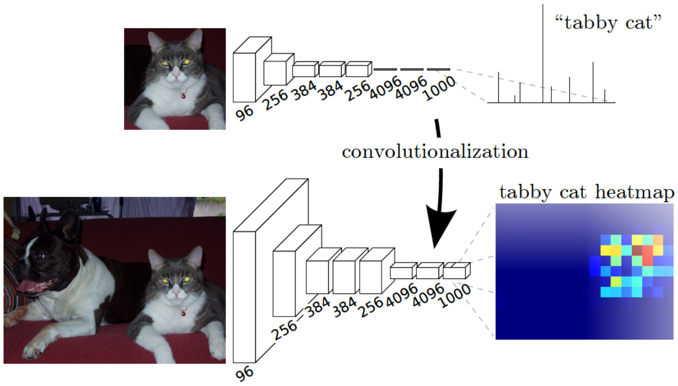


Figure 21.5: Transforming fully connected layers into convolution layers enables a classification net to output a heatmap.

[1] Long et al., *Fully Convolutional Networks for Semantic Segmentation*

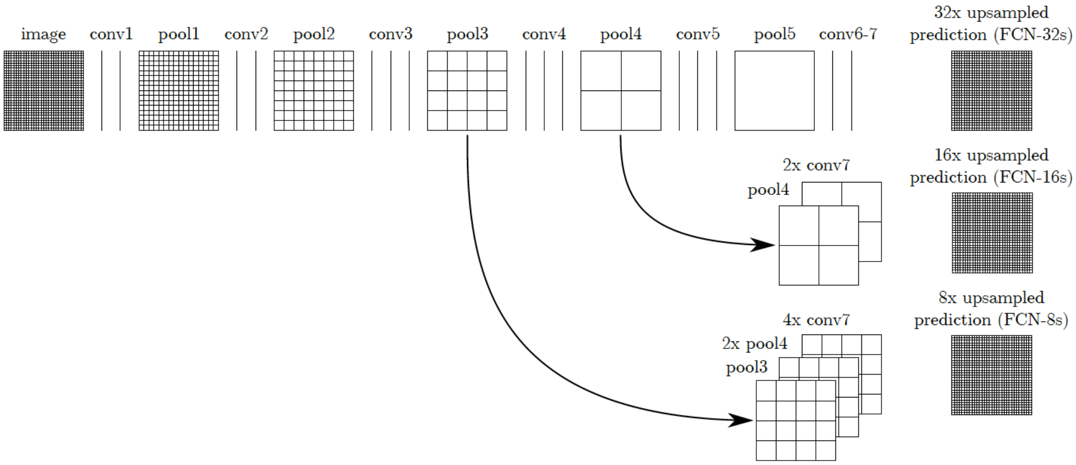


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure 21.6: A fully convolutional net (FCN) for segmentation that combines layers of the feature hierarchy and refines the spatial precision of the output.

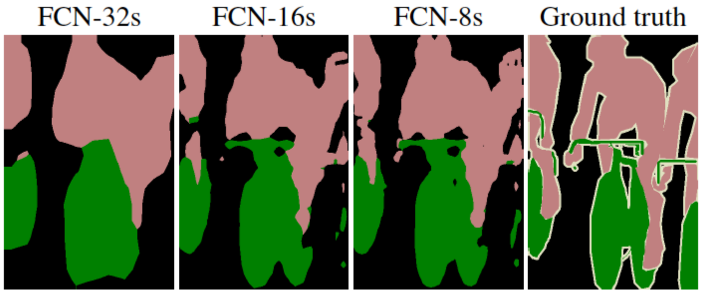


Figure 21.7: Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail.

2. U-Net [2]

   Since annotated examples are not as easily obtainable for biomedical tasks due to the significant and specialized effort required to procure them, U-Nets tries to use the provided samples efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization (see Figure 21.8). The contracting path involves repeatedly applying the following sequence: a 3x3 convolution, a ReLU, and a 2x2 max pool. The expansive step involves repeatedly applying the following sequence: upsampling the feature map followed by a 2x2 convolution ("”up-convolution”), a concatenation of the corresponding feature map from the contracting path, another 3x3 convolution, and a ReLU. At the final layer a 1x1 convolution is used to map each 64- component feature vector to the desired number of classes, for a total of 23 convolutional layers.

   For training, the energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function that penalizes deviation from the ground truth pixel labels (see Figure 21.9). These U-Nets train and run faster than the previous state-of-the art deep convolutional networks that came before it (10 hours to train, under a second to run), making them a compelling option. They are invariant to elastic deformations to the input image, a strategy that was leveraged to augment the data.

[2] Ronneberger et al., *U-Net: Convolutional Networks for Biomedical Image Segmentation*

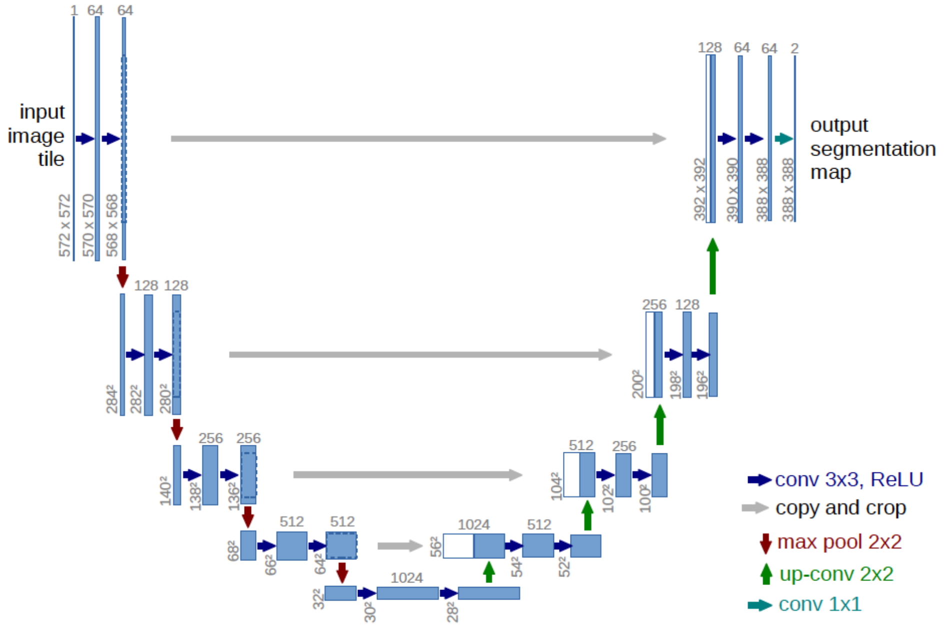
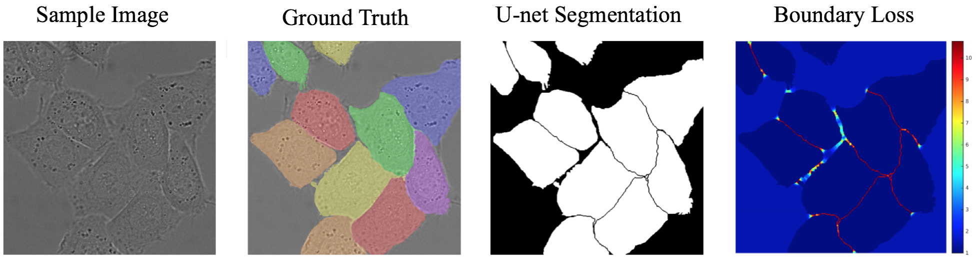


Figure 21.8: U-net architecture. Notice the namesake U shape.



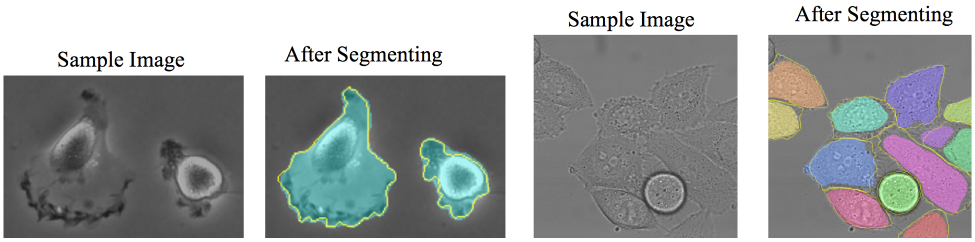
Sample Image

Ground Truth

U-net Segmentation

Boundary Loss

Figure 21.9: U-Net producing a pixel-wise loss map to force the network to learn border pixels



Sample Image

After Segmenting

Sample Image

After Segmenting

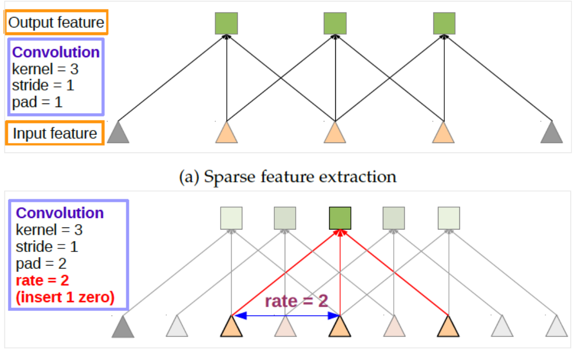
Figure 21.10: Example segmentation results with U-Net. The yellow border is the ground truth.

3. Atrous (Dilated) Convolution [3]

   Atrous convolution is convolution with upsampled filters. This allows us to explicitly control the resolution at which feature responses are computed within deep CNNs. Such an ability gives us a

[3] Chen et al., *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*

clear advantage: the output image will be closer in size to the input image, enabling sharper decision boundaries at useful scales. Traditional methods often involve downsampling steps that produce outputs significantly smaller than the inputs, and the corrective upsampling introduces pixel delocalization. To mitigate this, atrous convolutional networks combine the results at the final layer of a deep convolutional neural network with a fully connected Conditional Random Field (CRF), which is shown both qualitatively and quantitatively to improve localization performance.



(a) Sparse feature extraction

(b) Dense feature extraction

Figure 21.11: Atrous convolution in 1D

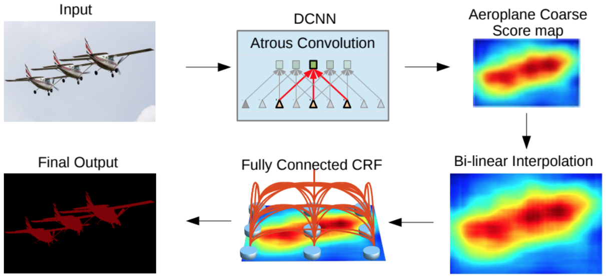


Figure 21.12: Overall procedure for atrous convolution

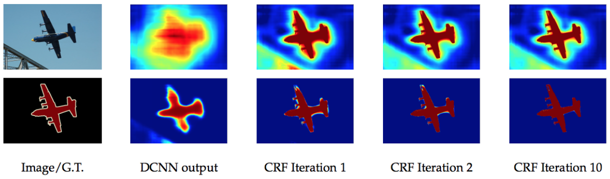


Figure 21.13: Repeated mean-field iterations further refine the score maps (top row) and belief maps (bottom row)

21.3 Healthcare Data Landscape and Diagnostic Radiology

21.3.1 Providers and Payers

In general, medical data is difficult to acquire. There are two primary repositories: healthcare providers and payers. Providers such as hospitals are often secretive about their datasets, which are often incomplete.

Payers such as insurance companies have more thorough datasets so long as the patient does not change insurers. Medicare data is somewhat perpetual as well but is biased towards older people.

21.3.2 Why Diagnostic Radiology is Important

1. Prevalence of studies: 600 million per year in the US, over 5 billion per year in the world

2. Shortage of skilled radiologists: 11,000 fewer than needed in the US, some countries only have a couple in total

3. Prevalence of errors: 5-15% error rate in the US, which has a very sophisticated medical system - this translates to 30-90 million misdiagnoses per year, a number of which are potentially fatal

21.3.3 AI in Diagnostic Radiology

Given an image along with possibly other modalities of data, the principal goals are to

1. Identify the region(s) of interest/anomaly

2. Infer whether the image has a clinically relevant medical condition

3. Infer the underlying cause of the condition (causal inference)

4. Write the full diagnostic report