Lecture 5: Medical Images -- Classification (cont.)
Announcements

- Upcoming deadlines:
  - A1 due next Wed Jan 29
  - Project proposal due Mon Feb 3
Last time: convolutional networks

32x32x3 image
5x5x3 filter

convolve (slide) over all spatial locations

activation maps

Slide credit: CS231n
ConvNet (or CNN) is a sequence of Convolution Layers, interspersed with activation functions.
**Evaluation metrics**

- **Receiver Operating Characteristic (ROC) curve:**
  - Plots sensitivity and specificity (specifically, 1 - specificity) as prediction threshold is varied
  - Gives trade-off between sensitivity and specificity
  - Also report summary statistic AUC (area under the curve)

![ROC curve example](image-url)
Evaluation metrics

- Selecting optimal trade-off points
  - **Youden’s Index**
    - \( J = \text{sensitivity} + \text{specificity} - 1 \)
    - Gives equal weight to optimizing true positives and true negatives
  - Sometimes also see F-measure (or F1 score)
    - \( F1 = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}} \)
    - Harmonic mean of precision and recall

But selected trade-off points could also depend on application

Also equal to distance above chance line for a balanced dataset: sensitivity - (1 - specificity) = sensitivity + specificity - 1

Figure credit: [https://en.wikipedia.org/wiki/File:ROC_Curve_Youden_J.png](https://en.wikipedia.org/wiki/File:ROC_Curve_Youden_J.png)
Today

- Continuation of convolutional networks for classification
- Medical image classification
- Start discussing segmentation and detection
LeNet-5

[LeCun et al., 1998]

Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-FC-FC]
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

- **2010**: 28.2
  - Lin et al

- **2011**: 25.8
  - Sanchez & Perronnin

- **2012**: 16.4
  - Krizhevsky et al (AlexNet)

- **2013**: 11.7
  - Krizhevsky & Fergus

- **2014**: 7.3
  - Simonyan & Zisserman (VGG)

- **2014**: 6.7
  - Szegedy et al (GoogLeNet)

- **2015**: 3.6
  - He et al (ResNet)

- **2016**: 3
  - Shao et al

- **2017**: 2.3
  - Hu et al (SENet)

- **Human**: 5.1

**First CNN-based winner**

- **8 layers**
  - 19 layers
  - 22 layers

- **152 layers**
  - 152 layers
  - 152 layers

*Slide credit: CS231n*
AlexNet

[Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

- **INPUT**: [227x227x3] input
- **CONV1**: 96 11x11 filters at stride 4, pad 0
- **MAX POOL1**: 3x3 filters at stride 2
- **NORM1**: Normalization layer
- **CONV2**: 256 5x5 filters at stride 1, pad 2
- **MAX POOL2**: 3x3 filters at stride 2
- **NORM2**: Normalization layer
- **CONV3**: 384 3x3 filters at stride 1, pad 1
- **CONV4**: 384 3x3 filters at stride 1, pad 1
- **CONV5**: 256 3x3 filters at stride 1, pad 1
- **MAX POOL3**: 3x3 filters at stride 2
- **FC6**: 4096 neurons
- **FC7**: 4096 neurons
- **FC8**: 1000 neurons (class scores)

Figure credit: Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.
AlexNet

[Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

- [227x227x3] INPUT
- [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
- [27x27x96] MAX POOL1: 3x3 filters at stride 2
- [27x27x96] NORM1: Normalization layer
- [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
- [13x13x256] MAX POOL2: 3x3 filters at stride 2
- [13x13x256] NORM2: Normalization layer
- [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
- [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
- [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
- [6x6x256] MAX POOL3: 3x3 filters at stride 2
- [4096] FC6: 4096 neurons
- [4096] FC7: 4096 neurons
- [1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission. Slide credit: CS231n
AlexNet

[Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

[227x227x3] INPUT

[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0

[27x27x96] MAX POOL1: 3x3 filters at stride 2

[27x27x96] NORM1: Normalization layer

[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2

[13x13x256] MAX POOL2: 3x3 filters at stride 2

[13x13x256] NORM2: Normalization layer

[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1

[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1

[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1

[6x6x256] MAX POOL3: 3x3 filters at stride 2

[4096] FC6: 4096 neurons

[4096] FC7: 4096 neurons

[1000] FC8: 1000 neurons (class scores)

Historical note: Trained on GTX 580 GPU with only 3 GB of memory. Network spread across 2 GPUs, half the neurons (feature maps) on each GPU.
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

- **2010**: Lin et al
- **2011**: Sanchez & Perronnin
- **2012**: Krizhevsky et al (AlexNet)
- **2013**: Zeiler & Fergus
- **2014**: Simonyan & Zisserman (VGG) (GoogLeNet)
- **2015**: He et al (ResNet)
- **2016**: Shao et al
- **2017**: Hu et al (SENet)
- **Human**

**Deeper Networks**

- 152 layers
- 152 layers
- 152 layers

**Shallow Networks**

- 8 layers
- 8 layers

**Layers**

- **2010**: 28.2
- **2011**: 25.8
- **2012**: 16.4
- **2013**: 11.7
- **2014**: 7.3
- **2014**: 6.7
- **2015**: 3.6
- **2016**: 3
- **2017**: 2.3
- **Human**: 5.1

*Slide credit: CS231n*
VGGNet

[Simonyan and Zisserman, 2014]

Small filters, Deeper networks

8 layers (AlexNet)
-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

11.7% top 5 error in ILSVRC’13 (ZFNet)
-> 7.3% top 5 error in ILSVRC’14

AlexNet  VGG16  VGG19

[Simonyan and Zisserman, 2014]
GoogLeNet
[Szegedy et al., 2014]

“Inception module”: design a good local network topology (network within a network) and then stack these modules on top of each other.
GoogLeNet
[ Szegedy et al., 2014 ]

Deeper networks, with computational efficiency

- 22 layers
- Efficient “Inception” module
- Avoids expensive FC layers
- 12x less params than AlexNet
- ILSVRC’14 classification winner (6.7% top 5 error)
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

“Revolution of Depth”

- 2010: Lin et al (shallow)
- 2011: Sanchez & Perronnin (AlexNet) (8 layers)
- 2012: Krizhevsky et al (8 layers)
- 2013: Zeiler & Fergus (19 layers)
- 2014: Simonyan & Zisserman (VGG) (22 layers)
- 2014: Szegedy et al (GoogLeNet)
- 2015: He et al (ResNet) (152 layers)
- 2016: Shao et al (3 layers)
- 2017: Hu et al (SENet) (152 layers)
- Human: 5.1

Slide credit: CS231n
ResNet

[He et al., 2015]

Very deep networks using residual connections

- 152-layer model for ImageNet
- ILSVRC’15 classification winner (3.57% top 5 error)
- Won all major classification and detection benchmark challenges in 2015
ResNet

[He et al., 2015]

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?

Q: What’s strange about these training and test curves? [Hint: look at the order of the curves]
ResNet

[He et al., 2015]

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?

56-layer model performs worse on both training and test error

-> The deeper model performs worse, but it’s not caused by overfitting!
ResNet

[He et al., 2015]

Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize
ResNet

[He et al., 2015]

Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize.

The deeper model should be able to perform at least as well as the shallower model.

A solution by construction is copying the learned layers from the shallower model and setting additional layers to identity mapping.
ResNet

[He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping

```
H(x)
```

```
conv
relu
conv
```

“Plain” layers

```
F(x) + x
```

```
relu
conv
relu
conv
```

Residual block

Slide credit: CS231n
ResNet

[He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping.

\[ H(x) = F(x) + x \]

“Plain” layers

Use layers to fit residual
\[ F(x) = H(x) - x \]
instead of
\[ H(x) \] directly

Residual block

Slide credit: CS231n
ResNet

[He et al., 2015]

Full ResNet architecture:
- Stack residual blocks
- Every residual block has two 3x3 conv layers

Slide credit: CS231n
ResNet

[He et al., 2015]

Full ResNet architecture:
- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)

Slide credit: CS231n
ResNet
[He et al., 2015]

Full ResNet architecture:
- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
ResNet

[He et al., 2015]

Full ResNet architecture:
- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)
ResNet

[He et al., 2015]

Total depths of 34, 50, 101, or 152 layers for ImageNet

Slide credit: CS231n
For deeper networks (ResNet-50+), use 1x1 “bottleneck” layer to improve efficiency (also used in GoogLeNet)
CNNs for Medical Imaging Classification
Early steps of deep learning in medical imaging: using ImageNet CNN features

Bar et al. 2015

- Input: Chest x-ray images
- Output: Several binary classification tasks
  - Right pleural effusion or not
  - Enlarged heart or not
  - Healthy or abnormal
- Very small dataset: 93 frontal chest x-ray images
Early steps of deep learning in medical imaging: using ImageNet CNN features

Bar et al. 2015

- Input: Chest x-ray images
- Output: Several binary classification tasks
  - Right pleural effusion or not
  - Enlarged heart or not
  - Healthy or abnormal
- Very small dataset: 93 frontal chest x-ray images

Early steps of deep learning in medical imaging: using ImageNet CNN features

Bar et al. 2015

- Input: Chest x-ray images
- Output: Several binary classification tasks
  - Right pleural effusion or not
  - Enlarged heart or not
  - Healthy or abnormal
- Very small dataset: 93 frontal chest x-ray images


Q: How might we approach this problem?
Bar et al. 2015

- Did not train a deep learning model on the medical data
- Instead, extracted features an AlexNet trained on ImageNet
  - 5th, 6th, and 7th layers
- Used features with an SVM (support vector machine) to perform classification
- Performed zero-mean unit-variance normalization of all features
- Evaluated combination with other hand-crafted image features (LBP, GIST, PiCoDes)

Q: How might we interpret the AUC vs. CNN feature trends?

Table 1. Right Pleural Effusion Condition.

<table>
<thead>
<tr>
<th></th>
<th>Low Level</th>
<th>High Level</th>
<th>Deep</th>
<th>Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LBP</td>
<td>GIST</td>
<td>PiCoDes</td>
<td>Decaf L5</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.71</td>
<td>0.79</td>
<td>0.79</td>
<td>0.93</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.77</td>
<td>0.92</td>
<td>0.91</td>
<td>0.84</td>
</tr>
<tr>
<td>AUC</td>
<td>0.75</td>
<td>0.93</td>
<td>0.91</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table 2. Healthy vs. Pathology.

<table>
<thead>
<tr>
<th></th>
<th>Low Level</th>
<th>High Level</th>
<th>Deep</th>
<th>Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LBP</td>
<td>GIST</td>
<td>PiCoDes</td>
<td>Decaf L5</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.65</td>
<td>0.68</td>
<td>0.59</td>
<td>0.73</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.61</td>
<td>0.66</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td>AUC</td>
<td>0.63</td>
<td>0.72</td>
<td>0.72</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 3. Enlarged Heart Condition.

<table>
<thead>
<tr>
<th></th>
<th>Low Level</th>
<th>High Level</th>
<th>Deep</th>
<th>Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LBP</td>
<td>GIST</td>
<td>PiCoDes</td>
<td>Decaf L5</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.75</td>
<td>0.79</td>
<td>0.79</td>
<td>0.88</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.78</td>
<td>0.81</td>
<td>0.84</td>
<td>0.78</td>
</tr>
<tr>
<td>AUC</td>
<td>0.80</td>
<td>0.82</td>
<td>0.87</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Ciompi et al. 2015

- Task: classification of lung nodules in 3D CT scans as peri-fissural nodules (PFN, likely to be benign) or not
- Dataset: 568 nodules from 1729 scans at a single institution. (65 typical PFNs, 19 atypical PFNs, 484 non-PFNs).
- Data pre-processing: prescaling from CT hounsfield units (HU) into [0,255]. Replicate 3x across R,G,B channels to match input dimensions of ImageNet-trained CNNs.
Ciompi et al. 2015

- Also extracted features from a deep learning model trained on ImageNet
  - Overfeat feature extractor (similar to AlexNet, but trained using additional losses for localization and detection)
  - To capture 3D information, extracted features from 3 different 2D views of each nodule, then input into 2-stage classifier (independent predictions on each view first, then outputs combined into second classifier).
Lakhani and Sundaram 2017

- Binary classification of pulmonary tuberculosis from x-rays
- Four de-identified datasets
- 1007 chest x-rays (68% train, 17.1% validation, 14.9% test)
- Now: training CNNs from scratch as well as fine-tuning from ImageNet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Untrained</th>
<th>Pretrained</th>
<th>Untrained with Augmentation*</th>
<th>Pretrained with Augmentation*</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.90 (0.84, 0.95)</td>
<td>0.98 (0.95, 1.00)</td>
<td>0.95 (0.90, 0.98)</td>
<td>0.98 (0.94, 0.99)</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>0.88 (0.81, 0.92)</td>
<td>0.97 (0.93, 0.99)</td>
<td>0.94 (0.89, 0.97)</td>
<td>0.98 (0.94, 1.00)</td>
</tr>
<tr>
<td>Ensemble</td>
<td></td>
<td></td>
<td>0.99 (0.96, 1.00)</td>
<td></td>
</tr>
</tbody>
</table>

Note.—Data in parentheses are 95% confidence interval.
* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

Lakhani and Sundaram 2017

- Binary classification of pulmonary tuberculosis from x-rays
- Four de-identified datasets
- 1007 chest x-rays (68% train, 17.1% validation, 14.9% test)
- Now: training CNNs from scratch as well as fine-tuning from ImageNet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Untrained</th>
<th>Pretrained</th>
<th>Untrained with Augmentation*</th>
<th>Pretrained with Augmentation*</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.90 (0.84, 0.95)</td>
<td>0.98 (0.95, 1.00)</td>
<td>0.95 (0.90, 0.98)</td>
<td>0.98 (0.94, 0.99)</td>
</tr>
<tr>
<td>GoogeLeNet</td>
<td>0.88 (0.81, 0.92)</td>
<td>0.97 (0.93, 0.99)</td>
<td>0.94 (0.89, 0.97)</td>
<td>0.98 (0.94, 1.00)</td>
</tr>
<tr>
<td>Ensemble</td>
<td></td>
<td></td>
<td></td>
<td>0.99 (0.96, 1.00)</td>
</tr>
</tbody>
</table>

Note.—Data in parentheses are 95% confidence interval.
* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

All training images were resized to 256x256 and underwent base data augmentation of random 227x227 cropping and mirror images. Additional data augmentation experiments in results table.

Lakhani and Sundaram 2017

- Binary classification of pulmonary tuberculosis from x-rays
- Four de-identified datasets
- 1007 chest x-rays (68% train, 17.1% validation, 14.9% test)
- Now: training CNNs from scratch as well as fine-tuning from ImageNet

All training images were resized to 256x256 and underwent base data augmentation of random 227x227 cropping and mirror images. Additional data augmentation experiments in results table.

<table>
<thead>
<tr>
<th>AUC Test Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>AlexNet</td>
</tr>
<tr>
<td>GoogLeNet</td>
</tr>
<tr>
<td>Ensemble</td>
</tr>
</tbody>
</table>

Note.—Data in parentheses are 95% confidence interval.

* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

Often resize to match input size of pre-trained networks. Also fine approach to making high-res dataset easier to work with!

Lakhani and Sundaram 2017

Performed further analysis at optimal threshold determined by the Youden Index.

Gulshan et al. 2016

- Binary classification of referable diabetic retinopathy from retinal fundus photographs (moderate and worse diabetic retinopathy, referable diabetic macular edema, or both)
- Huge data curation effort! 128,175 images, each graded by 3-7 ophthalmologists.
  - 54 total graders, each paid to grade between 20 to 62508 images.
- Inception-v3 (GoogLeNet) CNN with ImageNet pre-training

Gulshan et al. 2016

AUC = 0.991

Looked at different operating points
- High-specificity point approximated ophthalmologist specificity for comparison. Should also use high-specificity to make decisions about high-risk actions.
- High-sensitivity point should be used for screening applications.

Gulshan et al. 2016

Q: What could explain the difference in trends for reducing # grades / image on training set vs. tuning set, on tuning set performance?

Esteva et al. 2017

- Two binary classification tasks: malignant vs. benign lesions of epidermal or melanocytic origin
- Inception-v3 (GoogLeNet) CNN with ImageNet pre-training
- Fine-tuned on dataset of 129,450 lesions (from several sources) comprising 2,032 diseases
- Evaluated model vs. 21 or more dermatologists in various settings

Esteva et al. 2017

- Train on finer-grained classification (757 classes) but perform binary classification at inference time by summing probabilities of fine-grained sub-classes
- The stronger fine-grained supervision during the training stage improves inference performance!
Esteva et al. 2017

- Evaluation of algorithm vs. dermatologists

Rajpurkar et al. 2017

- Binary classification of pneumonia presence in chest X-rays
- Used ChestX-ray14 dataset with over 100,000 frontal X-ray images with 14 diseases
- 121-layer DenseNet CNN
- Compared algorithm performance with 4 radiologists
- Also applied algorithm to other diseases to surpass previous state-of-the-art on ChestX-ray14

Wu et al. 2019

- Binary classification of breast malignant and benign findings
- Model based on ResNet architecture
- Multi-view network (preview of multimodal models!)

Wu et al. 2019

- Binary classification of breast malignant and benign findings
- Model based on ResNet architecture
- Multi-view network (preview of multimodal models!)

Wu et al. 2019

- Binary classification of breast malignant and benign findings
- Model based on ResNet architecture
- Multi-view network (preview of multimodal models!)

McKinney et al. 2020

- Binary classification of breast cancer in mammograms
- International dataset and evaluation, across UK and US

Richer visual recognition tasks: segmentation and detection

**Classification**
Output: one category label for image (e.g., colorectal glands)

**Semantic Segmentation**
Output: category label for each pixel in the image

**Detection**
Output: Spatial bounding box for each instance of a category object in the image

**Instance Segmentation**
Output: Category label and instance label for each pixel in the image

Richer visual recognition tasks: segmentation and detection

**Classification**
- Output: one category label for image (e.g., colorectal glands)

**Semantic Segmentation**
- Output: category label for each pixel in the image

**Detection**
- Output: Spatial bounding box for each **instance** of a category object in the image

**Instance Segmentation**
- Output: Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

Semantic segmentation: U-Net

Semantic segmentation: U-Net

Semantic segmentation: U-Net

Semantic segmentation: U-Net


Output is an image mask: width x height x # classes

Output image size a little smaller than original, due to convolutional operations w/o padding

Gives more “true” context for reasoning over each image area. Can tile to make predictions for arbitrarily large images
Semantic segmentation: U-Net

Max pooling enables aggregation of increasingly more context (higher level features)

Semantic segmentation: U-Net

Semantic segmentation: U-Net

Semantic segmentation: U-Net


Up-convolutions to go from the global information encoded in highest-level features, back to individual pixel predictions.
Up-convolutions

Recall: Normal 3 x 3 convolution, stride 2 pad 1

Input: 4 x 4

Output: 2 x 2
Up-convolutions

Recall: Normal 3 x 3 convolution, stride 2 pad 1

Input: 4 x 4

Dot product between filter and input

Output: 2 x 2
Up-convolutions

Recall: Normal 3 x 3 convolution, **stride 2** pad 1

Input: 4 x 4

Dot product between filter and input

Output: 2 x 2

Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output
Up-convolutions

3 x 3 transpose convolution, stride 2 pad 1

Input: 2 x 2

Output: 4 x 4
Up-convolutions

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

Input: 2 x 2

Output: 4 x 4

Input gives weight for filter
Up-convolutions

3 x 3 transpose convolution, stride 2 pad 1

Input: 2 x 2

Output: 4 x 4

Input gives weight for filter

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input
Up-convolutions

3 x 3 transpose convolution, stride 2 pad 1

Input gives weight for filter

Input: 2 x 2

Output: 4 x 4

Sum where output overlaps

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input
Up-convolutions

Other names:
- Transpose convolution
- Fractionally strided convolution
- Backward strided convolution

3 x 3 transpose convolution, stride 2 pad 1

Input gives weight for filter

Input: 2 x 2

Output: 4 x 4

Sum where output overlaps

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Up-convolutions
 Semantic segmentation: U-Net

Semantic segmentation: U-Net

Train with classification loss (e.g. binary cross entropy) on every pixel, sum over all pixels to get total loss

Semantic segmentation: IOU evaluation

Intersection over Union:

\[ IoU = \frac{target \cap prediction}{target \cup prediction} \]

- \# pixels included in both target and prediction maps
- Total \# pixels in the union of both masks
Semantic segmentation: IOU evaluation

**Intersection over Union:**

\[
\text{IoU} = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}}
\]

Can compute this over all masks in the evaluation set, or at individual mask and image levels to get finer-grained understanding of performance.
Semantic segmentation: U-Net cell segmentation

<table>
<thead>
<tr>
<th>Name</th>
<th>PhC-U373</th>
<th>DIC-HeLa</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMCB-SG (2014)</td>
<td>0.2669</td>
<td>0.2935</td>
</tr>
<tr>
<td>KTH-SE (2014)</td>
<td>0.7953</td>
<td>0.4607</td>
</tr>
<tr>
<td>HOUS-US (2014)</td>
<td>0.5323</td>
<td>-</td>
</tr>
<tr>
<td>second-best 2015</td>
<td>0.83</td>
<td>0.46</td>
</tr>
<tr>
<td>u-net (2015)</td>
<td><strong>0.9203</strong></td>
<td><strong>0.7756</strong></td>
</tr>
</tbody>
</table>

Richer visual recognition tasks: segmentation and detection

**Classification**

Output: one category label for image (e.g., colorectal glands)

**Semantic Segmentation**

Output: category label for each pixel in the image

**Detection**

Output: Spatial bounding box for each instance of a category object in the image

**Instance Segmentation**

Output: Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

Object detection: Faster R-CNN

CNN backbone (any CNN network that produces spatial feature map outputs)
Object detection: Faster R-CNN

Regress to bounding box “candidates” from “anchor boxes” at each location

Classification loss
Bounding-box regression loss
Rol pooling
proposals
Region Proposal Network
feature map

256-d intermediate layer
2k scores cls layer
4k coordinates reg layer
k anchor boxes

sliding window
conv feature map

CNN

image
Object detection: Faster R-CNN

In each of top bounding box candidate locations, crop features within box (treat as own image) and perform further refinement of bounding box + classification.
Cropping Features: RoI Pool

Divide into grid of (roughly) equal subregions, corresponding to fixed-size input required for final classification / bounding box regression networks.

“Snap” to grid cells.

Max-pool within each subregion.

Richer visual recognition tasks: segmentation and detection

**Classification**

Output: one category label for image (e.g., colorectal glands)

**Semantic Segmentation**

Output: category label for each pixel in the image

**Detection**

Output: Spatial bounding box for each instance of a category object in the image

**Instance Segmentation**

Output: Category label and instance label for each pixel in the image


Distinguishes between different instances of an object.
Instance segmentation: Mask R-CNN

Add a small mask network that operates on each RoI to predict a segmentation mask.
Cropping Features: RoI Align

Sample at regular points in each subregion using bilinear interpolation

No “snapping”!

Improved version of RoI Pool since we now care about pixel-level segmentation accuracy!

Image features (e.g. 512 x 20 x 15)
Cropping Features: RoI \textit{Align}

Improved version of RoI Pool since we now care about pixel-level segmentation accuracy!

Feature $f_{xy}$ for point $(x, y)$ is a linear combination of features at its four neighboring grid cells.

No “snapping”!

Sample at regular points in each subregion using bilinear interpolation.

Image features
Summary

Today we saw:

- Concrete formulations of widely used CNN architectures
- Some case studies of CNNs for medical image classification
- Started discussing segmentation and detection

Next time:

- More on segmentation and detection for medical images