Lecture 6: Structured Prediction with ConvNets

Deep Learning @ UvA
Previous Lecture

- What do convolutions look like?
- Build on the visual intuition behind Convnets
- Deep Learning Feature maps
- Transfer Learning
Lecture Overview

- What is structured prediction?
- Can we repurpose for structured prediction?
- Structured losses on ConvNets
- Multi-task learning with ConvNets
What is structured prediction?
Standard inference

- N-way classification
Standard inference

- N-way classification
- Regression

How popular will this movie be in IMDB?
Standard inference

- N-way classification
- Regression
- Ranking
- ...

Who is older?
What do they have in common?

- Do all our machine learning tasks boil to “single value” predictions?
- Are there tasks where outputs are somehow correlated?
- Is there some structure in this output correlations?
- How can we predict such structures? \(\rightarrow\) Structured prediction
What do they have in common?

- They all make “single value” predictions
- Do all our machine learning tasks boil down to “single value” predictions?
Object detection

- Predict a box around an object
  - Images
    - Spatial location $\rightarrow$ b(ounding) box
  - Videos
    - Spatio-temporal location $\rightarrow$ bbox@$t$, bbox@$t+1$, ...

![Image of object detection](image)
Object Segmentation

Image | Class map | Instance map | Part map | Part map (high level)
Optical flow & motion estimation

(a) Consecutive frames
(b) Trajectories from Optical Flow
(c) \( \omega \)-trajectories
Depth estimation

Input left  Ours stereo  Ours mono

Godard et al., Unsupervised Monocular Depth Estimation with Left-Right Consistency, 2016
Normals and reflectance estimation

*Wang et al., Designing deep networks for surface normal estimation, 2015*

*Rematas et al., Deep Reflectance Maps, 2016*
Sentence parsing
Machine translation

Economic growth has slowed down in recent years.

Das Wirtschaftswachstum hat sich in den letzten Jahren verlangsamt.

La croissance économique s'est ralenti ces dernières années.
And many more

- Speech synthesis
- Captioning
- Robot control
- Pose estimation
- ...

...
What is common?

- Prediction goes beyond asking for “single values”
- Outputs are complex and output dimensions correlated
Structured prediction

- Prediction goes beyond asking for “single values”
- Outputs are complex and output dimensions correlated
- Output dimensions have latent structure
- Can we make deep networks to return **structured predictions?**
Structured prediction

- Prediction goes beyond asking for “single values”
- Outputs are complex and output dimensions correlated
- Output dimensions have latent structure
- Can we make deep networks to return structured predictions?
ConvNets for structured prediction
Sliding window on feature maps

- SPPnet [He2014]
- Fast R-CNN [Girshick2015]
Fast R-CNN: Steps

- Process the whole image up to conv5
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects (some correct, most wrong)
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects
- Given single location $\rightarrow$ ROI pooling module extracts fixed length feature

Conv 5 feature map

Always 4x4 no matter the size of candidate location
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects
- Given single location $\rightarrow$ ROI pooling module extracts fixed length feature
- Connect to two final layers, 1 for classification, 1 for box refinement

Conv 1 $\rightarrow$ Conv 2 $\rightarrow$ Conv 3 $\rightarrow$ Conv 4 $\rightarrow$ Conv 5 feature map

ROI Pooling Module

Always $3 \times \#$ no matter the size of candidate location

Car, dog or bicycle?

New box coordinates
Region-of-Interest (ROI) Pooling Module

- Divide feature map in $T \times T$ cells
  - The cell size will change depending on the size of the candidate location

Always 3x3 no matter the size of candidate location
Smart fine-tuning

- Normally samples in a mini-batch completely random
- Instead, organize mini-batches by ROIs
- 1 mini-batch = $N$ (images) $\times \frac{R}{N}$ (candidate locations)
- Feature maps shared $\rightarrow$ training speed-up by a factor of $\frac{R}{N}$
- Mini-batch samples might be correlated
  - In Fast R-CNN that was not observed
Some results
Fast-RCNN

- Reuse convolutions for different candidate boxes
  - Compute feature maps only once

- Region-of-Interest pooling
  - Define stride relatively → box width divided by predefined number of “poolings” \( T \)
  - Fixed length vector

- End-to-end training!
  - (Very) Accurate object detection
  - (Very) Faster
    - Less than a second per image
  - External box proposals needed
Faster R-CNN [Girshick2016]

- Fast R-CNN $\rightarrow$ external candidate locations
- Faster R-CNN $\rightarrow$ deep network proposes candidate locations
- Slide the feature map $\rightarrow$ $k$ anchor boxes per slide

Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the ‘attention’ of this unified network.
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input $\Rightarrow$ slide the network

``
\text{Is this pixel a camel?}
\begin{tabular}{c|c}
Yes! & No! \\
\end{tabular}
``
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input → slide the network

Is this pixel a camel? • Yes!  • No!
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input $\Rightarrow$ slide the network

Is this pixel a camel? 
- Yes! 
- No!
Image larger than network input → slide the network
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input $\Rightarrow$ slide the network

Is this pixel a camel?
- Yes!  
- No!
Connect intermediate layers to output

Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.
Fully Convolutional Networks [LongCVPR2014]

- Output is too coarse
  - Image Size 500x500, Alexnet Input Size: 227x227 → Output: 10x10

- How to obtain dense predictions?

- Upconvolution
  - Other names: deconvolution, transposed convolution, fractionally-strided convolutions
Deconvolutional modules

Convolution
No padding, no strides

Upconvolution
No padding, no strides

Upconvolution
Padding, strides

More visualizations: https://github.com/vdumoulin/conv_arithmetic
Coarse $\rightarrow$ Fine Output

Small loss generated

Large loss generated (probability much higher than ground truth)

Ground truth pixel labels

Pixel label probabilities

Upconvolution 2x

Upconvolution 2x

7x7

14x14

224x224

Pixel label probabilities:

0.8 0.1 0.9

[49x9]UVA DEEP LEARNING COURSE – EFSTRATIOS GAVVES

[928x9] STRUCTURED PREDICTION WITH CONVNETS - 41
Structured losses
Deep ConvNets with CRF loss [Chen, Papandreou 2016]

- Segmentation map is good but not pixel-precise
  - Details around boundaries are lost
- Cast fully convolutional outputs as unary potentials
- Consider pairwise potentials between output dimensions
Deep ConvNets with CRF loss [Chen, Papandreou 2016]
Deep ConvNets with CRF loss [Chen, Papandreou 2016]

- Segmentation map is good but not pixel-precise
  - Details around boundaries are lost
- Cast fully convolutional outputs as unary potentials
- Consider pairwise potentials between output dimensions
- Include Fully Connected CRF loss to refine segmentation

$$E(x) = \sum \theta_i(x_i) + \sum \theta_{ij}(x_i, x_j)$$

<table>
<thead>
<tr>
<th>Total loss</th>
<th>Unary loss</th>
<th>Pairwise loss</th>
</tr>
</thead>
</table>

$$\theta_{ij}(x_i, x_j) \sim w_1 \exp \left( -\alpha |p_i - p_j|^2 - \beta |I_i - I_j|^2 \right) + w_2 \exp(-\gamma |p_i - p_j|^2)$$
Deep ConvNets with CRF loss: Examples
One image ➔ Several tasks

- Per image we can predict, boundaries, segmentation, detection, ...
  - Why separately?
- Solve multiple tasks simultaneously
- One task might help learn another better
- One task might have more annotations
- In real applications we don’t want 7 VGGnets
  - 1 for boundaries, 1 for normals, 1 for saliency, ...
Multi-task learning

- The total loss is the summation of the per task losses
- The per task loss relies on the common weights (VGGnet) and the weights specialized for the task

\[ \mathcal{L}_{total} = \sum_{task} \mathcal{L}_{task}(\theta_{common}, \theta_{task}) + \mathcal{R}(\theta_{task}) \]

- One training image might contain specific only annotations
  - Only a particular task is “run” for that image
- Gradients per image are computed for tasks available for the image only
Ubernet [Kokkinos2016]
Naïve backpropagation

Figure 5: Vanilla backpropagation for multi-task training: a naive implementation has a memory complexity $2N(L_C + TL_T)$, where here $L_C = 6$ is the depth of the common CNN trunk, $L_T = 3$ is the depth of the task-specific branches and $T = 2$ is the number of tasks.
Question

- So far, what have you noticed?
Question

- So far, what have you noticed?
- Most works are done in 2016
  - Very fast, very active, very volatile area of research that attracts lots of interest
Summary

- What is structured prediction?
- Can we repurpose for structured prediction?
- Structured losses on ConvNets
- Multi-task learning with ConvNets
Reading material & references


http://www.deeplearningbook.org/

Part III: Chapter 16
Next lecture

- Deep Learning and Natural Language
- Invited lecture given by Prof. Christof Monz