C14200: FROM ZERO TO ONE - DEEP LEARNING WITH PYTORCH

JOE SPISAK
PRODUCT MANAGER

FRANCISCO MASSA
RESEARCH ENGINEER
WHO AM I?

CURRENT: RESEARCH ENGINEER - PYTORCH

PREVIOUS:
- COMPUTER VISION RESEARCH ENGINEER @TWITTER

- PHD STUDENT AT ECOLE DES PONTS - FRANCE
AGENDA

01
PRIMER ON COMPUTER VISION

02
GOING BEYOND IMAGE CLASSIFICATION

03
COMPUTER VISION WITH TORCHVISION

04
HANDS ON WITH DEEPLAB V3

05
HANDS ON WITH KEYPOINT R-CNN
PRIMER ON COMPUTER VISION
How can we make the computer understand what is in the image?

Image from https://www.researchgate.net/publication/330902210/figure/fig1/AS:878026619375622@1586349267376/image-of-Abraham-Lincoln-as-a-matrix-of-pixel-values.ppm
OBJECTS AS A BAG OF VISUAL WORDS

Prototype patches (visual words)

Histogram of visual words

Image from https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb
ALEXNET - A REVOLUTION IN COMPUTER VISION

ImageNet Image Classification Challenge

1M Images
1000 categories

Image from https://medium.com/@atharvabarve24/breakthrough-in-computer-vision-c318feda4d2
ALEXNET - A REVOLUTION IN COMPUTER VISION

ImageNet Image Classification Challenge

Ranking of the best results from each team

Key ingredients:
- Deep Convolutional Neural Networks
- Lot's of training data
- ReLU and dropout
- GPUs

Image from https://medium.com/@atharvabarve24/breakthrough-in-computer-vision-c318fedba4d2
What is a convolution?

```
<table>
<thead>
<tr>
<th>7</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>
```

```
7x1+4x1+3x1+  
2x0+5x0+3x0+  
3x-1+3x-1+2x-1  
= 6
```

Image from https://medium.com/datadriveninvestor/convolutional-neural-networks-3b241a5da51e
Neural networks that uses the convolution operator

Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.
CONVOLUTIONAL NEURAL NETWORKS TODAY

More layers work better

Image from https://medium.com/@pierre_guillou/understand-how-works-resnet-without-talking-about-residual-64698f157e0c
GOING BEYOND IMAGE CLASSIFICATION
WHAT IS IN THE IMAGE, AND WHERE?

Classification

Classification + Localization

Object Detection

Instance Segmentation

Semantic Segmentation

---

Images from https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852 and https://medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7
CNN OVER REGIONS (R-CNN)

WHAT IS TORCHVISION?

A library built to facilitate research and experimentation in the field of Computer Vision
<table>
<thead>
<tr>
<th>Datasets</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Datasets</td>
<td>Pre-Trained Models</td>
</tr>
<tr>
<td>Ops</td>
<td>Transforms</td>
</tr>
<tr>
<td>Efficient Operators</td>
<td>Data Transformation</td>
</tr>
<tr>
<td>Io</td>
<td>References</td>
</tr>
<tr>
<td>Efficient Video Reader</td>
<td>Training Scripts</td>
</tr>
</tbody>
</table>
import torchvision.datasets

torchvision.datasets.MNIST(root='.', download=True)

torchvision.datasets.CIFAR10(root='.', download=True)

torchvision.datasets.CelebA(root='.', download=True)

torchvision.datasets.ImageNet('.
', download=True)
import torchvision.transforms as T
T.Compose([T.RandomResizedCrop(224), T.RandomHorizontalFlip(), T.ToTensor(), T.Normalize(mean=[0.486, 0.456, 0.406], std=[0.229, 0.224, 0.225])])
An Analysis of Deep Neural Network Models for Practical Applications
Alfredo Canziani, Adam Paszke, Eugenio Culurciello
import torchvision.ops

torchvision.ops.box_iou(...)
torchvision.ops.roi_align(...)
torchvision.ops.nms(...)
torchvision.ops.roi_pool(...)
import torchvision.io

torchvision.io.read_video(filename, start_pts=0, end_pts=None)

torchvision.io.read_video_timestamps(filename)

torchvision.io.write_video(filename, video_array, fps, video_codec='libx264', options=None)
PRE-TRAINED MODELS
WHAT IS TORCHVISION?

SIMPLIFY RESEARCH AND EXPERIMENTATION IN COMPUTER VISION

KeypointRCNN

MaskRCNN

DeepLabV3
import torchvision

model = torchvision.models.detection.maskrcnn_resnet50_fpn(pretrained=True)

# set it to evaluation mode, as the model behaves differently
# during training and evaluation
model.eval()

image = PIL.Image.open('/path/to/an/image.jpg')
image_tensor = torchvision.transforms.functional.to_tensor(image)

# pass a list of (potentially different sized) tensors
# to the model, in 0-1 range. The model will take care of
# batching them together and normalizing
output = model([image_tensor])

# output is a list of dict, containing the post processed predictions
HANDS ON WITH TORCHVISION