AGENDA

0. Quick recap
1. Deep Learning
2. Computer Vision:
   a. Convolutional Neural Network (CNN)
   b. Applications and Examples
3. Transfer Learning
QUICK RECAP
Supervised Learning

\[ X \rightarrow \text{Model} \rightarrow Y \]

\[ \hat{y}^{(i)} = f_\theta(x^{(i)}) \]

Gradient to the rescue!

\[ \theta^{(i+1)} = \theta^{(i)} - \alpha \cdot \nabla L(\theta) \]
DEEP LEARNING
DEEP LEARNING

AI APPLICATIONS

Source: https://www.slideshare.net/NVIDIA/deep-learning-workflows-training-and-inference
COMPUTER VISION
IMAGE CLASSIFICATION

Assign correct label set to an input image:

- CAT?
- DOG?
- DUCK?
IMAGE CLASSIFICATION

We could use Fully-Connected Neural Networks (FCNN) from last time...
NEURAL NETWORKS FOR CV

... but they probably won’t work well. Why is that?

- Waaaaaaaaay too much parameters...
- Not good with spatial data...
CONVOLUTIONAL NEURAL NETWORKS

Adapted from last year’s slides on Computer Vision by Jakub Powierza
CNN ARCHITECTURE
CONVOLUTION

- Creates "feature maps",
- Apply filters on the image,
- Move such filter over the image and calculate feature,
- Follow the stride (how many fields it should "jump"),
- Is defined by kernel size (filter size),
- Can use padding for bigger receptive field.
CONVOLUTION

Image

Convolved Feature
STRIDE

Stride = 1

Stride = 2
PADDDING

Padding = 0

Padding = 2
RECEPTIVE FIELD
POOLING

- Pooling reduces spatial space,
- Reduces amount of parameters,
- Reduces overfitting,
- A simple routing (during back propagation),
- Most common: MaxPooling,
- Also: AvgPooling, ...
SAMPLE ARCHITECTURE: VGG

224 × 224 × 3  224 × 224 × 64

112 × 112 × 128

56 × 56 × 256

28 × 28 × 512

14 × 14 × 512

1 × 1 × 4096  1 × 1 × 1000

- convolution + ReLU
- max pooling
- fully connected + ReLU
- softmax
COMPUTER VISION APPLICATIONS
IMAGE CLASSIFICATION

Assign correct label set to an input image:
**IMAGE CLASSIFICATION**

**High accuracy:**
- Inception, Xception
- ResNet, ResNext
- DenseNet
- NasNet
- SENet

**Fast:**
- MobileNet
- ShuffleNet

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-152 [10]</td>
<td>23.0</td>
<td>6.7</td>
<td>21.3</td>
<td>5.5</td>
</tr>
<tr>
<td>Inception-v3 [44]</td>
<td>-</td>
<td>-</td>
<td>21.2</td>
<td>5.6</td>
</tr>
<tr>
<td>Inception-v4 [42]</td>
<td>-</td>
<td>-</td>
<td>20.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Inception-ResNet-v2 [42]</td>
<td>-</td>
<td>-</td>
<td>19.9</td>
<td>4.9</td>
</tr>
<tr>
<td>ResNeXt-101 (64 × 4d) [47]</td>
<td>20.4</td>
<td>5.3</td>
<td>19.1</td>
<td>4.4</td>
</tr>
<tr>
<td>DenseNet-264 [14]</td>
<td>22.15</td>
<td>6.12</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Attention-92 [46]</td>
<td>-</td>
<td>-</td>
<td>19.5</td>
<td>4.8</td>
</tr>
<tr>
<td>Very Deep PolyNet [51]</td>
<td>-</td>
<td>-</td>
<td>18.71</td>
<td>4.25</td>
</tr>
<tr>
<td>PyramidNet-200 [8]</td>
<td>20.1</td>
<td>5.4</td>
<td>19.2</td>
<td>4.7</td>
</tr>
<tr>
<td>SENet-154</td>
<td>18.68</td>
<td>4.47</td>
<td>17.28</td>
<td>3.79</td>
</tr>
<tr>
<td>NASNet-A (6@4032) [55]</td>
<td>-</td>
<td>-</td>
<td>17.3²</td>
<td>3.8²</td>
</tr>
<tr>
<td>SENet-154 (post-challenge)</td>
<td>-</td>
<td>-</td>
<td>16.88²</td>
<td>3.58²</td>
</tr>
</tbody>
</table>

State-of-the-art CNNs results on ImageNet validation set.

PROBLEM

What if we need infer real values from images instead of predicting a class?

REGRESSION

- Can be solved just like multilabel classification
- Use proper final activation function e.g. sigmoid for $<0,1>$ outputs or no activation
- Change loss function
PROBLEM

What if images could belong to multiple classes at once

MULTILABEL CLASSIFICATION

▪ Can be solved just like multiclass classification
▪ Change final activation function: Softmax → Sigmoid
▪ Change loss function
▪ Watch out! Class balancing and data splits become a challenge!

Image source: https://www.slideshare.net/dloiacono/ahmadi
PROBLEM

Train a model for predicting variables when instead of single images with labels we are given bags of images, such that the label refers to some image or images in the bag.

MULTIPLE INSTANCE LEARNING

▪ If we can predict variables on bag-level, we can treat entire bags as images or aggregate image features

▪ See https://www.kaggle.com/c/yelp-restaurant-photo-classification as an example

▪ If we must predict variable on image level, the problem is more difficult. We’ll come back to this later on.

Image source:
PROBLEM

Classify image and predict localization of an object

Image source:
OBJECT LOCALIZATION

Classification + Localization

Treat localization as a regression problem!

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Correct box:
(x', y', w', h')

L2 Loss

Box Coordinates
(x, y, w, h)

Fully Connected:
4096 to 1000

Vector:
4096

Correct label:
Cat

Image source:
PROBLEM

Detect multiple objects of the same kind with potential occlusion

Image source: https://www.microsoft.com/developerblog/2017/04/10/end-end-object-detection-box/
OBJECT DETECTION

- Accurate, 2-stage detectors:
  - Faster-RCNN
  - ... and its numerous extensions

- Fast and also accurate 1-stage detectors:
  - YOLO
  - SSD
  - RetinaNet
OBJECT DETECTION

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP-50</th>
<th>mAP-100</th>
<th>mAP-1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD321</td>
<td>48.1</td>
<td>48.1</td>
<td>48.1</td>
</tr>
<tr>
<td>R-FCN</td>
<td>51.9</td>
<td>51.9</td>
<td>51.9</td>
</tr>
<tr>
<td>DSSD513</td>
<td>53.3</td>
<td>53.3</td>
<td>53.3</td>
</tr>
<tr>
<td>FPN FRCN</td>
<td>59.1</td>
<td>59.1</td>
<td>59.1</td>
</tr>
<tr>
<td>RetinaNet-50</td>
<td>50.9</td>
<td>50.9</td>
<td>50.9</td>
</tr>
<tr>
<td>RetinaNet-101-500</td>
<td>53.1</td>
<td>53.1</td>
<td>53.1</td>
</tr>
<tr>
<td>RetinaNet-101-800</td>
<td>57.5</td>
<td>57.5</td>
<td>57.5</td>
</tr>
<tr>
<td>YOLOv3-320</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
</tr>
<tr>
<td>YOLOv3-416</td>
<td>55.3</td>
<td>55.3</td>
<td>55.3</td>
</tr>
<tr>
<td>YOLOv3-608</td>
<td>57.9</td>
<td>57.9</td>
<td>57.9</td>
</tr>
</tbody>
</table>
OBJECT DETECTION

Great open-source code:

- **TensorFlow Object Detection API**
  https://github.com/tensorflow/models/tree/master/research/object_detection
  • Faster R-CNN, SSD, Mask R-CNN

- **Facebook Detectron (for Caffe2)**
  https://github.com/facebookresearch/Detectron
  • RetinaNet, Faster R-CNN, Mask R-CNN

- **YOLO v3**
  • Yolo v1, v2, v3
PROBLEM

Generate text description for an image

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."
IMAGE CAPTIONING

Sequence prediction: CNN + LSTM
PROBLEM

Read text from an image

Image source:
http://teaching.paganstudio.com/digitalfoundations/?p=171
OPTICAL CHARACTER RECOGNITION

- Sequence prediction again: CNN + LSTM
- Attention models again
- Example: Google’s Attention-OCR

PROBLEM

What if we need to locate object but there are no locations in the training data?

LOCATION = ???!!!
WEAKLY-SUPERVISED OBJECT LOCALIZATION

Class Activation Maps

CheXNet: Pneumonia Detection on Chest X-Rays
WEAKLY-SUPERVISED OBJECT LOCALIZATION

Grad-CAM and guided grad-CAM activation and saliency maps

PROBLEM

Partition an image into meaningful regions

Image source:
https://vision.in.tum.de/research/image_segmentation
SEMANTIC SEGMENTATION

Fully Convolutional Networks

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input: $3 \times H \times W$

High-res: $D_1 \times H/2 \times W/2$

Low-res: $D_3 \times H/4 \times W/4$

Med-res: $D_2 \times H/4 \times W/4$

High-res: $D_1 \times H/2 \times W/2$

Predictions: $H \times W$

Image source:
SEMANTIC SEGMENTATION

U-Net networks

- Commonly used architecture
- Performs well on low-size datasets
- Multiple applications in medical image processing research
- Unfortunately, quite slow

Image source: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/
PROBLEM

Find similar images for given image query
IMAGE RETRIEVAL / VISUAL SEARCH

Example: Pinterest Visual Search

- AlexNet and VGG bottlenecks
- “salient color signatures” (segmentation, color clustering)
- Object detection using text information

THERE IS MUCH MORE...

- Face recognition
- Generative models
- 3D reconstruction
- Pose estimation
- Odometry
- Object tracking
- ...

TRANSFER LEARNING
HANDS-ON
TRANSFER LEARNING

1. Train on Imagenet

```
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
```

2. Small dataset: feature extractor

```
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
```

Freeze these

3. Medium dataset: finetuning

```
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
```

Train this

more data = retrain more of the network (or all of it)

Freeze these

Source: http://cs231n.github.io/
TRANSFER LEARNING

Not only Computer Vision...

OpenAI Retro Contest
April 5 to June 5, 2018

Source: OpenAI Retro Contest Results

Word embeddings: Male-Female and Verb tense

Source: word2vec paper

Source: Universal Language Model Fine-tuning for Text Classification
HANDS-ON

ImageNet

Transfer Learning

<table>
<thead>
<tr>
<th>Location</th>
<th>DSFL</th>
<th>DeCAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>auditorium</td>
<td>66.70%</td>
<td>38.89%</td>
</tr>
<tr>
<td>corridor</td>
<td>57.14%</td>
<td>47.62%</td>
</tr>
<tr>
<td>bowling</td>
<td>75.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>winecellar</td>
<td>23.81%</td>
<td>76.19%</td>
</tr>
</tbody>
</table>

MIT Indoor 67
ResNet-152

Revolution of Depth

Source: ResNet paper
TRANSFER LEARNING in KERAS

```python
from keras.applications import VGG16

vgg_conv = VGG16(weights='imagenet',
                 include_top=False,
                 input_shape=(224, 224, 3))
```

Transfer learning is super easy in Keras, if you use pretrained models available in `keras.applications`.

ResNet-152 isn’t there, so we will use another way to load pretrained model.
LET’S START THE HANDS-ON!

Repository:  
https://github.com/gberinger/resnet-finetune-demo

Original code:  
https://github.com/cta-ai/resnet-finetune-demo

Tutorial describing the code in detail (in Polish):  
READING MATERIALS
What to do next?

- **Stanford CS231n** - Convolutional Neural Networks for Visual Recognition:
  - Website - lecture notes etc.
  - YT lectures with Andrey Karpathy

- **Deep Learning Specialization** on Coursera with Andrew Ng
  - Specialization Info
  - Course 4: Convolutional Neural Networks

- Welch Labs - Learning to See

- A curated list of Awesome Computer Vision Resources

- Read links from this presentation if you’re interested in particular application!
Thank you!