Similarity learning
What can ML do for us?

• Classification problem
What can ML do for us?

• Classification problem on ImageNet with thousands of categories
What can ML do for us?

- Performance on ImageNet
  - Size of the blobs indicates the number of parameters

What can ML do for us?

- Regression problem: pose regression
What can ML do for us?

- Regression problem: bounding box regression

D. Held et al. „Learning to Track at 100 FPS with Deep Regression Networks“. ECCV 2016
What can ML do for us?

• Third type of problems

A  Classification: person, face, female

B  Classification: person, face, male
What can ML do for us?

• Third type of problems

Is it the same person?

A

B

Prof. Leal-Taixé and Prof. Niessner
What can ML do for us?

- Third type of problems: Similarity Learning
  
  - Comparison
  - Ranking
Similarity Learning: when and why?

- Application: unlocking your iPhone with your face

Training
Similarity Learning: when and why?

- Application: unlocking your iPhone with your face

Testing:

Can be solved as a classification problem
Similarity Learning: when and why?

- Application: face recognition system so students can enter the exam room without the need for ID check
Similarity Learning: when and why?

• Application: face recognition system so students can enter the exam room without the need for ID check

What is the problem with this approach?

Scalability – we need to retrain our model every time a new student is registered to the course
Similarity Learning: when and why?

- Application: face recognition system so students can enter the exam room without the need for ID check

Can we train one model and use it every year?
Similarity Learning: when and why?

• Learn a similarity function

A

Low similarity score

B

A

High similarity score

B

Prof. Leal-Taixé and Prof. Niessner
Similarity Learning: when and why?

• Learn a similarity function

Low similarity score

High similarity score
Similarity Learning: when and why?

- Learn a similarity function: testing

\[ d(A, B) > \tau \]

Not the same person
Similarity Learning: when and why?

- Learn a similarity function

\[ d(A, B) < \tau \]

Same person
Similarity learning

• How do we train a network to learn similarity?
Siamese Neural Networks
Similarity learning

• How do we train a network to learn similarity?

A → CNN → FC → Representation of my face in 128 values

Prof. Leal-Taixé and Prof. Niessner

Taigman et al. „DeepFace: closing the gap to human level performance“. CVPR 2014
How do we train a network to learn similarity?

A → f(A)

B → f(B)

Taigman et al. „DeepFace: closing the gap to human level performance“. CVPR 2014
Similarity learning

- Siamese network = shared weights

Taigman et al. „DeepFace: closing the gap to human level performance“. CVPR 2014
Similarity learning

• Siamese network = shared weights

• We use the same network to obtain an encoding of the image $f(A)$

• To be done: compare the encodings
Similarity learning

- Distance function \( d(A, B) = \| f(A) - f(B) \|^2 \)

- Training: learn the parameter such that
  - If \( A \) and \( B \) depict the same person, \( d(A, B) \) is small
  - If \( A \) and \( B \) depict a different person, \( d(A, B) \) is large
Similarity learning

• Loss function for a positive pair:

  – If $A$ and $B$ depict the same person, $d(A, B)$ is small

  \[
  \mathcal{L}(A, B) = \|f(A) - f(B)\|^2
  \]
Similarity learning

• Loss function for a negative pair:
  
  – If $A$ and $B$ depict a different person, $d(A, B)$ is large

  – Better use a Hinge loss

$$\mathcal{L}(A, B) = \max(0, m^2 - \|f(A) - f(B)\|^2)$$
Similarity learning

- Contrastive loss:

\[ \mathcal{L}(A, B) = y^* \| f(A) - f(B) \|^2 + (1 - y^*) \max(0, m^2 - \| f(A) - f(B) \|^2) \]

Positive pair, reduce the distance between the elements

Negative pair, brings the elements further apart up to a margin
Similarity learning

• Training the siamese networks
  – You can update the weights for each channel independently and then average them

• This loss function allows us to learn to bring positive pairs together and negative pairs apart
Triplet loss

- Triplet loss allows us to learn a ranking

We want: \[ \| f(A) - f(P) \|^2 < \| f(A) - f(N) \|^2 \]

Schroff et al. „FaceNet: a unified embedding for face recognition and clustering“. CVPR 2015
Triplet loss

- Triplet loss allows us to learn a ranking

\[ ||f(A) - f(P)||^2 < ||f(A) - f(N)||^2 \]
\[ ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 < 0 \]
\[ ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m < 0 \]

margin

Triplet loss

- Triplet loss allows us to learn a ranking

\[ \|f(A) - f(P)\|^2 < \|f(A) - f(N)\|^2 \]
\[ \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 < 0 \]
\[ \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + m < 0 \]

\[ L(A, P, N) = \max(0, \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + m) \]

Schroff et al „FaceNet: a unified embedding for face recognition and clustering“. CVPR 2015
Triplet loss

- Training with hard cases

\[ \mathcal{L}(A, P, N) = \max(0, \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + m) \]

- Train for a few epoch
- Choose the hard cases where \( d(A, P) \approx d(A, N) \)
- Train with those to refine the distance learned
Applications in vision
Establishing image correspondences

Image from University of Washington
Establishing image correspondences
Establishing image correspondences

• Used in a wide range of Computer Vision applications
  – Image stitching or image alignment
  – Object recognition
  – 3D reconstruction
  – Object tracking
  – Image retrieval

• Many of these applications are now targeted directly with Neural Networks as we will see in the course
Establishing image correspondences

• Classic method pipeline
  – Extract manually designed feature descriptors
    • Harris, SIFT, SURF: most are based on image gradients
    • They suffer under extreme illumination or viewpoint changes
    • Slow to extract dense features
  – Match descriptors from the two images
    • Many descriptors are similar, one needs to filter out possible double matches and keep only reliable ones.

Sameer Agarwal et al. „Building Rome in a Day“. ICCV 2009
Establishing image correspondences

- End-to-end learning for patch similarity
- Fast to allow dense extraction
- Invariant to a wide array of transformations (illumination, viewpoint)

S. Zagoruyko and N. Komodakis. „Learning to Compare Image Patches via Convolutional Neural Networks“. CVPR 2015
Establishing image correspondences

- Classic Siamese architecture
  - Shared layers
    - Simulated feature extraction
  - One decision layer
    - Simulates the matching

S. Zagoruyko and N. Komodakis. „Learning to Compare Image Patches via Convolutional Neural Networks“. CVPR 2015
Image retrieval

Radenovic et al. „Fine-tuning CNN Image Retrieval with No Human Annotation“. TPAMI 2018
Unsupervised learning

- Learning from videos
  - Tracking provides the supervision
  - Use those as positive samples
  - Extract random patches as negative samples

Optical flow

• Input: 2 consecutive images (e.g. from a video)
• Output: displacement of every pixel from image A to image B

• Results in the “perceived” 2D motion, not the real motion of the object
Optical flow
Optical flow
Optical flow with CNNs

- End-to-end supervised learning of optical flow

Optical flow with CNNs

FlowNet: Learning Optical Flow with Convolutional Networks

We train convolutional networks to estimate optical flow.

FlowNet: architecture 1

- Stack both images → input is now 2 x RGB = 6 channels
FlowNet: architecture 2

• Siamese architecture
FlowNet: architecture 2

- Two key design choices

How to combine the information from both images?
Correlation layer

• Multiplies a feature vector with another feature vector

Fixed operation. No learnable weights!
Correlation layer

- The matching score represents how correlated these two feature vectors are
Correlation layer

• Useful for finding image correspondences

Correlation layer

FlowNet: architecture 2

- Two key design choices

How to combine the information from both images?

How to obtain high-quality results?
FlowNet: architecture 2

- Convolutions + pooling are great to allow aggregation of information from different parts of the image.

- It also makes computation feasible!

- Problem: it reduces the size of our input, if we want full sized outputs (segmentation, optical flow) we need further operations.
Refinement architecture
Transpose convolution

• Recall

Convolution
no padding, no stride

Convolution
padding, stride
Transpose convolution

- We want to convert the 3x3 input into a 5x5 output
- Clever padding on the input plus a normal convolution
- Unpooling + conv = upconvolution
More on that later

• Next step: Autoencoder architecture as to generate outputs of the same size as inputs
Cool things you can do

• Savinov et al. „Quad-networks: unsupervised learning to rank for interest point detection“. CVPR 2017

• Ristani & Tomasi. „Features for Multi-Target Multi-Camera Tracking and Re-Identification“. CVPR 2018

• Chen et al. „Beyond triplet loss: a deep quadruplet network for person re-identification“. CVPR 2017
Next lecture

• No session on Friday

• Next Monday: more on advanced architectures