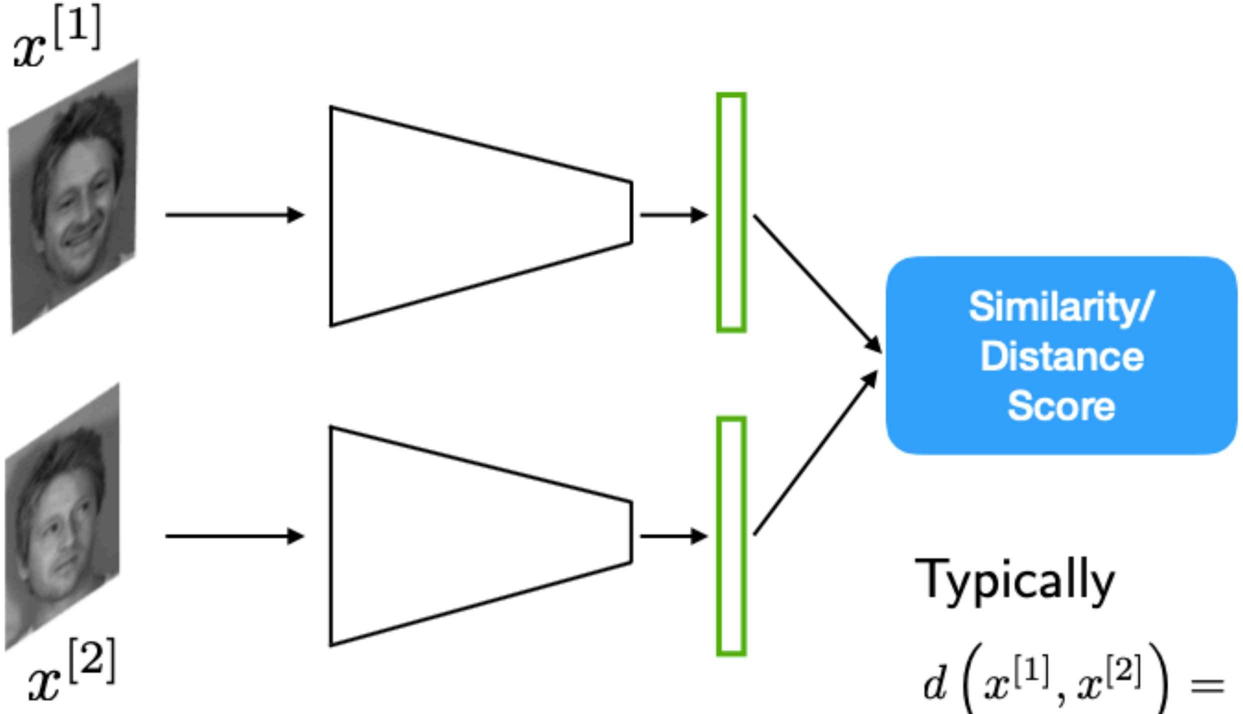
Face Recognition

Suleyman Demirel University
CSS634: Deep Learning
PhD Abay Nussipbekov

Face Recognition and Metric Learning

Siamese Networks



Typically

$$d\left(x^{[1]}, x^{[2]}\right) = \left\|f\left(x^{[1]}\right) - f\left(x^{[1]}\right)\right\|_2^2$$

or

$$d\left(x^{[1]}, x^{[2]}\right) = \left|f\left(x^{[1]}\right) - f\left(x^{[1]}\right)\right|_1$$

Siamese Networks

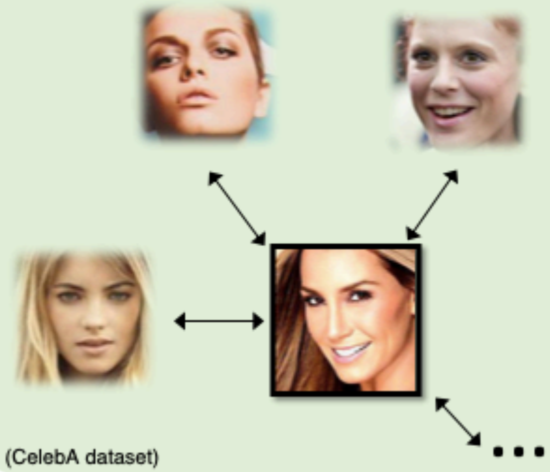
Often used for "One-shot learning"

- Suppose you trained a Siamese network for verification tasks
- Now, suppose you have only ~1 object per class
- You can compare any new object to any object based on maximum similarity to your given images (somewhat related to K-nearest neighbors)

Face Recognition:
Face Identification vs Face Verification

A. Identification

Determine identity of an unknown person
1-to-*n* matching



(CelebA dataset)

B. Verification

Verify claimed identity of a person
1-to-1 matching

(MUCT dataset)

dataset link: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

dataset link: http://www.milbo.org/muct/

FaceNet - Face Verification

Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "[Facenet: A unified embedding for face recognition and clustering]." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.

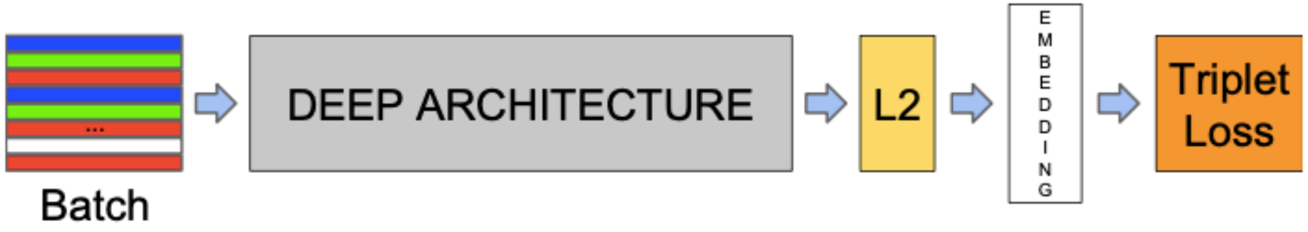


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

FaceNet - Face Verification

Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.

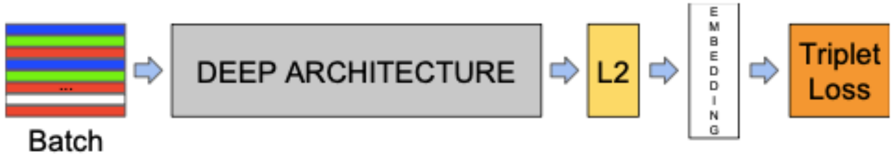


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

Triplet Loss

Anchor Positive

Want encodings to be very similar
(small distance)

Anchor Negative

Want encodings to be very different
(large distance)

Triplet Loss

Anchor Positive

Want encodings to be very similar
(small distance)

Anchor Negative

Want encodings to be very different
(large distance)

$$d(A, P) \leq d(A, N)$$

$$\|f(A) - f(P)\|_2^2 \leq \|f(A) - f(N)\|_2^2$$

Triplet Loss


Anchor Positive

Want encodings to be very similar
(small distance)


Anchor Negative

Want encodings to be very different
(large distance)

$$d(A, P) + \alpha \leq d(A, N)$$

$$\|f(A) - f(P)\|_2^2 + \boxed{\alpha} \leq \|f(A) - f(N)\|_2^2$$

To make it a little harder

Triplet Loss

Anchor Positive

Want encodings to be very similar
(small distance)

Anchor Negative

Want encodings to be very different
(large distance)

$$d(A, P) + \alpha \leq d(A, N)$$

$$\|f(A) - f(P)\|_2^2 + \alpha \leq \|f(A) - f(N)\|_2^2$$

Rearrange

$$\boxed{\|f(A) - f(P)\|_2^2 + \alpha - \|f(A) - f(N)\|_2^2 \leq 0}$$

Triplet Loss

Anchor Positive

Want encodings to be very similar
(small distance)

Anchor Negative

Want encodings to be very different
(large distance)

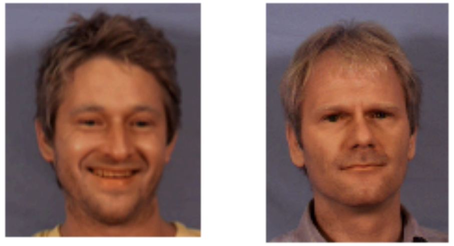
Bounded loss function for training:

$$\mathcal{L}(A, P, N) = \max\left(\|f(A) - f(P)\|_2^2 + \alpha - \|f(A) - f(N)\|_2^2, 0\right)$$

Triplet Loss

Anchor Positive

Want encodings to be very similar
(small distance)



Anchor Negative

Want encodings to be very different
(large distance)

<u>In practice:</u> Selecting good pairs (those that are "hard") is crucial during training

$$\mathcal{L}(A, P, N) = \max\left(\|f(A) - f(P)\|_2^2 + \alpha - \|f(A) - f(N)\|_2^2, 0\right)$$

Optional: Recent Triplet Loss Variants

(not required), only for those who are interested

- Cosine Similarity-based triplet loss:

Li, Chao, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. "Deep speaker: an end-to-end neural speaker embedding system." *arXiv preprint arXiv:1705.02304* (2017).

- Angular Loss:

Wang, Jian, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. "Deep metric learning with angular loss." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2593-2601. 2017.

- Large margin cosine loss:

Wang, Hao, Yitong Wang, Zheng` Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. "Cosface: Large margin cosine loss for deep face recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265-5274. 2018.

Resources Used

- Deeplearning.ai by Andre Ng
- STAT 479: Deep Learning by Sebastian Raschka