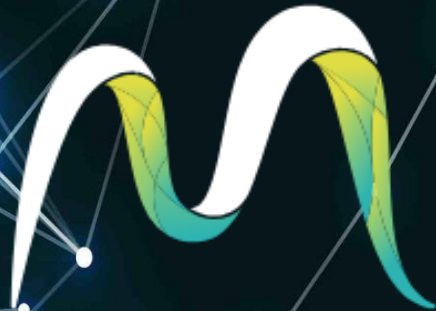


metaliquid



Nicolò Gregori,
Senior Data Scientist
mail: nicolo@meta-liquid.com



metaliquid

Website: www.meta-liquid.com

Blog: <https://medium.com/metaliquid>



METALIQUID



AI video content analysis startup

designed for the media and broadcast industry delivers
timecoded descriptive metadata from digital media assets






VIDEO CONTENT IS EVERYWHERE

Metaliquid analyzes video content with **human intelligence quality**,
but **faster** and in **real time**.

Metaliquid recognizes **concepts** in videos and identifies their **meaning** and **relation**.
Our neural nets can be trained to identify **specific features upon request**.



PART I

THEORY

IMAGE CLASSIFICATION

IMAGE CLASSIFICATION: the task is to predict a *semantic* label given an input image.

WHO IS THIS PERSON?

Sofia Loren

Uma Thurman

Leonardo di Caprio

George Clooney

Kate Winslet



Emilia Clarke

Brad Pitt

Cristiano Ronaldo

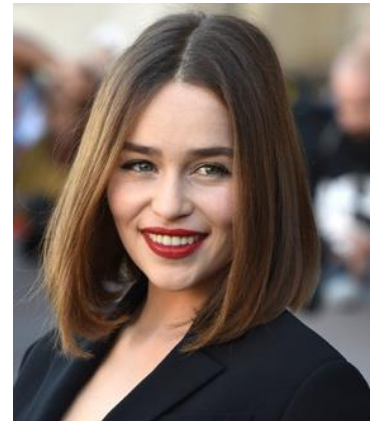
Orlando Bloom

Demi Moore

IMAGE VERIFICATION

IMAGE VERIFICATION: the task is to assign a probability indicating whether two images are *close together*.

ARE THEY THE SAME PERSON?



What is meant by *close together* ?

The objective is to find a way to measure how much two images are

SEMANTICALLY close

IMAGE CLASSIFICATION

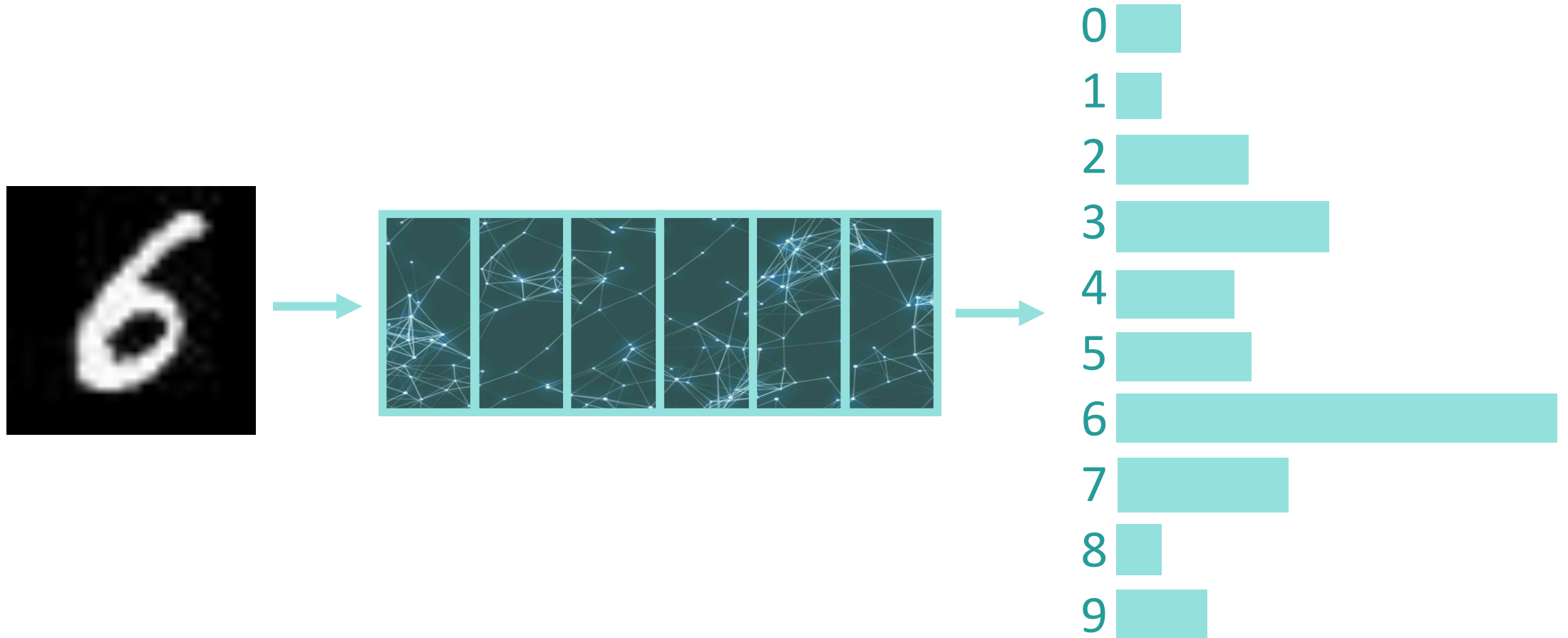
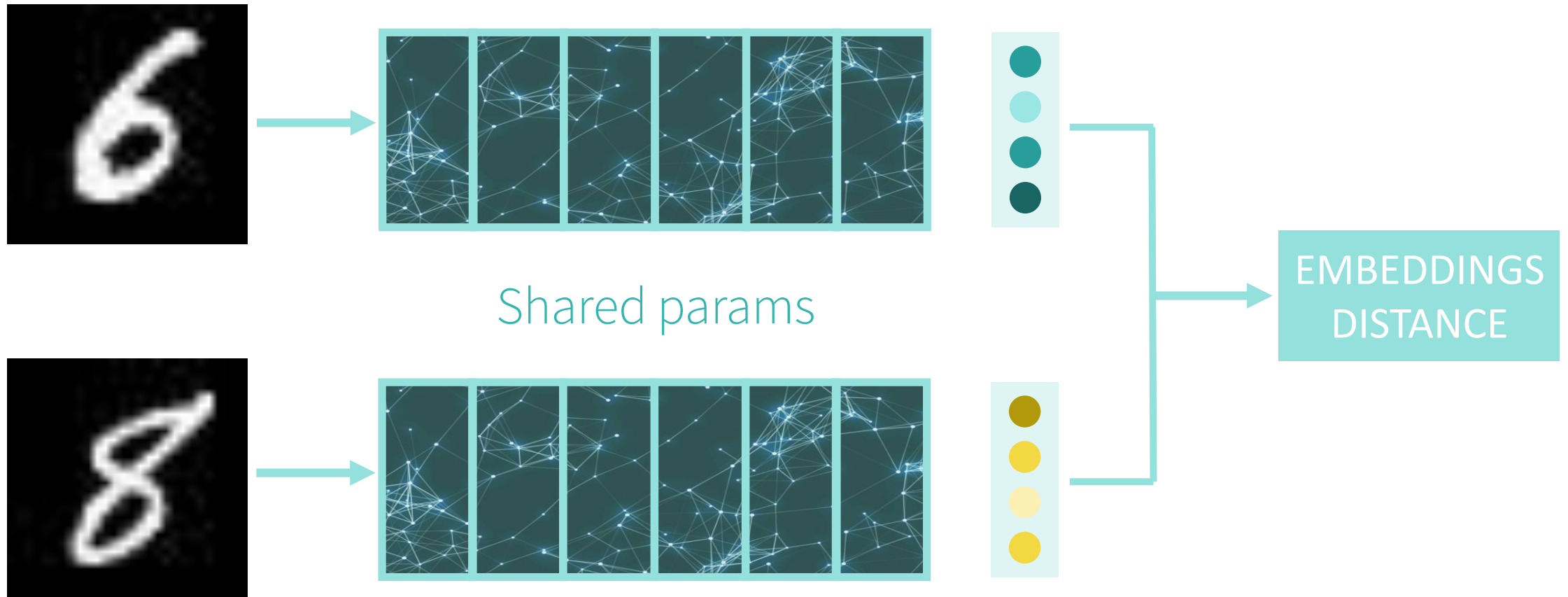


IMAGE VERIFICATION



IMAGE VERIFICATION



One shot learning

Image dissimilarity

ONE SHOT LEARNING

Train a very deep convolutional neural network require a **huge amount of data**.
One shot learning aims to assign the correct semantic label given a **few example images**.

SIAMESE NEURAL NETWORK

can achive the task to understand if two images belong to the same category,
so it is possible to decide if a new image represents the same concept of other
images in the dataset.



?



IMAGE DISSIMILARITY

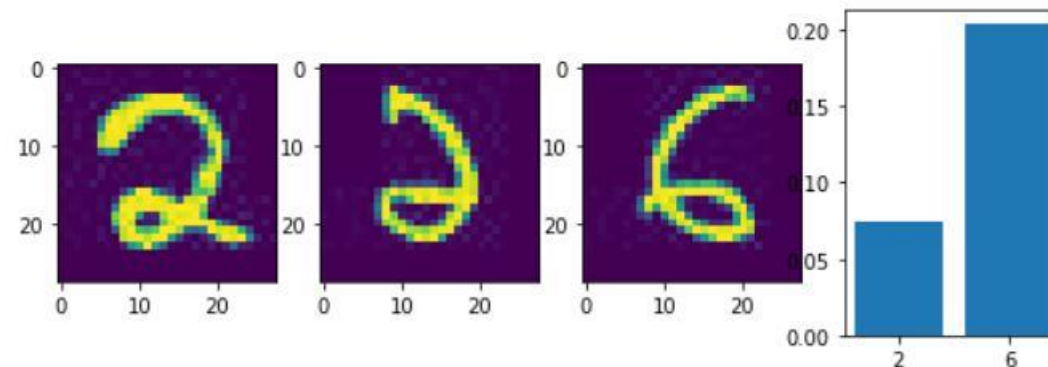
Some tasks require to understand if two images are similar or not (e.g. shot detection), but deterministic models are not always appropriate.

For example, the following images are very similar in the pixel space



SIAMESE NEURAL NETWORK

Can fit a «semantic» dissimilarity function between two images.



A LITTLE DEEPER ...

Learning a Similarity Metric Discriminatively, with Application to Face Verification

Sumit Chopra

Raia Hadsell

Yann LeCun

Courant Institute of Mathematical Sciences
New York University
New York, NY, USA
{sumit, raia, yann}@cs.nyu.edu

Abstract

We present a method for training a similarity metric from data. The method can be used for recognition or verification applications where the number of categories is very large and not known during training, and where the number of training samples for a single category is very small. The idea is to learn a function that maps input patterns into a target space such that the L_1 norm in the target space approximates the "semantic" distance in the input space. The

per category. A common approach to this kind of problem is distance-based methods, which consist in computing a similarity metric between the pattern to be classified or verified and a library of stored prototypes. Another common approach is to use non-discriminative (generative) probabilistic methods in a reduced-dimension space, where the model for one category can be trained without using examples from other categories. To apply discriminative learning techniques to this kind of application, we must devise a method that can extract information about the problem

FaceNet: A Unified Embedding for Face Recognition and Clustering

Florian Schroff
fschroff@google.com
Google Inc.

Dmitry Kalenichenko
dkalenichenko@google.com
Google Inc.

James Philbin
jphilbin@google.com
Google Inc.

Abstract

Despite significant recent advances in the field of face recognition [10, 14, 15, 17], implementing face verification and recognition efficiently at scale presents serious challenges to current approaches. In this paper we present a system, called FaceNet, that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors.

Our method uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning

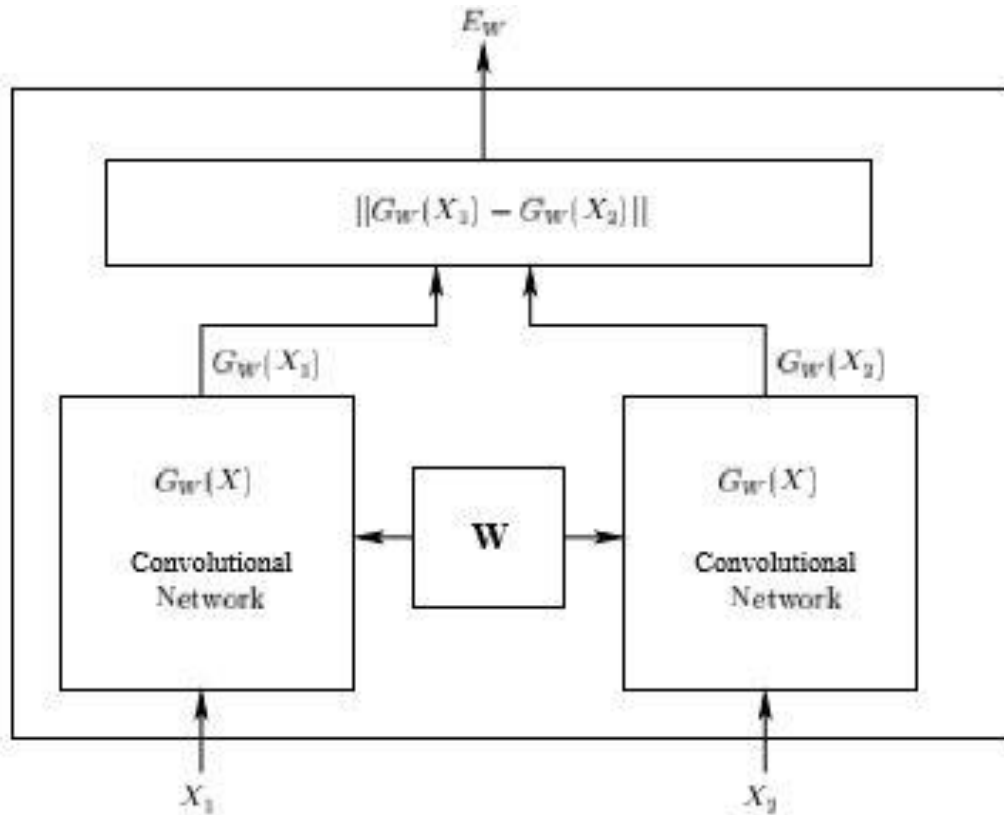


[cs.CV] 17 Jun 2015

LEARNING A SIMILARITY METRIC DISCRIMINATIVELY: APPLICATION TO FACE VERIFICATION

- **Authors:** Sumit Chopra, Raia Hadsell and Yann LeCun(Turing Award 2018 WINNER).
- **Year:** 2005.
- **Main Idea:** « find a function that maps input patterns into a target space such that a simple distance in the target space (say the Euclidean distance) approximates the “semantic” distance in the input space ».
- **Architecture:** Siamese.
- **Cost function to optimize:** Contrastive loss function.

LEARNING A SIMILARITY METRIC DISCRIMINATIVELY: APPLICATION TO FACE VERIFICATION



- Complete freedom to choose $G_W(X)$ (vgg, resnet, ecc)
- $E_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||$
- $E_W(X_1, X_2) = E_W(X_2, X_1)$
- **Objective:** given a family of functions $G_W(X)$ parameterized by W and the similarity metric $E_W(X_1, X_2)$. The objective is to minimize E_W when X_1 and X_2 are from the same category and maximize E_W when X_1 and X_2 belong to different category.

LEARNING A SIMILARITY METRIC DISCRIMINATIVELY: APPLICATION TO FACE VERIFICATION

- $E_w(X_1, X_2) = ||G_w(X_1) - G_w(X_2)||$
- Let E_w^g (genuine) if X_1 *and* X_2 belong the same category and E_w^i (impostor) if they represent different concepts.
- Condition: $\exists m > 0, \text{ such that } E_w^g + m < E_w^i$. The positive number m can be interpreted as a margin
- $Y=0$, $Y=1$ (Same , Different)

LEARNING A SIMILARITY METRIC DISCRIMINATIVELY: APPLICATION TO FACE VERIFICATION

- Contrastive Loss
- Calculated on two input images

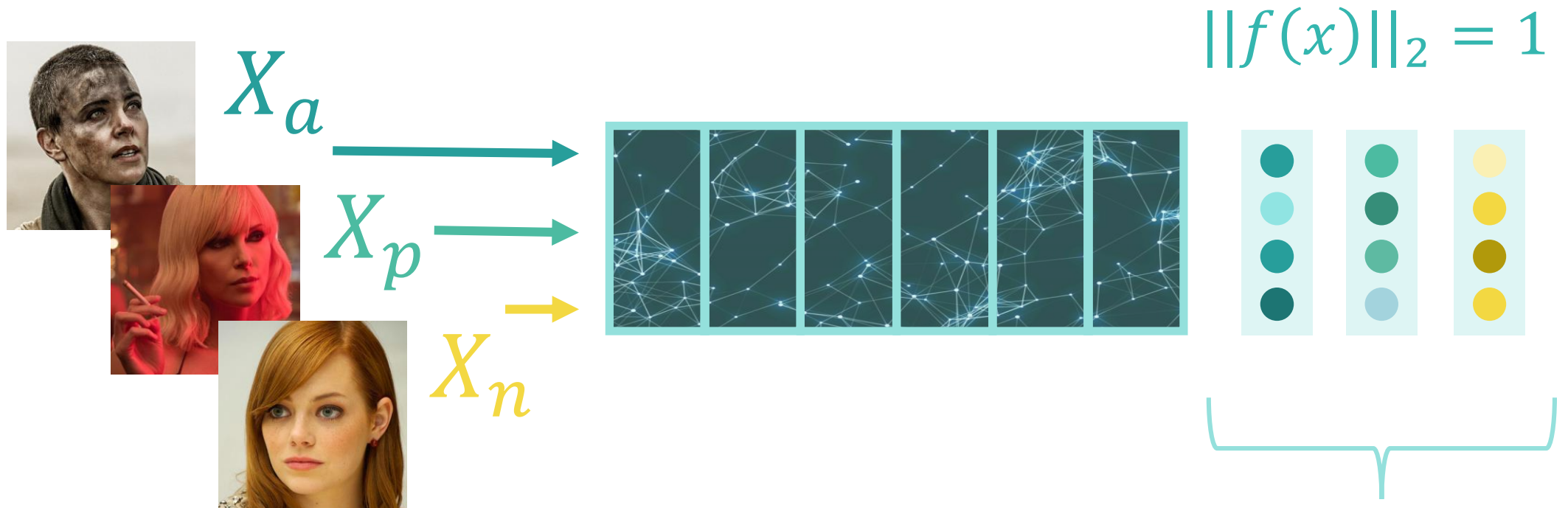
$$L(X_1, X_2) = (1 - Y) \frac{1}{2} (E_w)^2 + Y \frac{1}{2} \{\max(0, m - E_w)\}^2$$

$$L(X_1, X_2) = \begin{cases} \frac{1}{2} \{\max(0, m - E_w)\}^2 & \text{if } X_1, X_2 \text{ not belong the same category} \\ \frac{1}{2} (E_w)^2 & \text{if } X_1, X_2 \text{ belong the same category} \end{cases}$$

FACENET: A UNIFIED EMBEDDING FOR FACE RECOGNITION AND CLUSTERING

- **Authors:** Florian Schroff, Dmitry Kalenichenko and James Philbin.
- **Year:** 2015 (ten years later ...).
- **Main Ideas:**
 - « the squared distance between *all* faces, independent of imaging conditions, of the same identity is small, whereas the squared distance between a pair of face images from different identities is large »;
 - « Generating all possible triplets would result in many triplets that are easily satisfied. [...] It is crucial to select hard triplets, that are active and can therefore contribute to improving the model ».
- **Architecture:** Siamese.
- **Cost function to optimize:** Triplet loss function.

IMAGE VERIFICATION



TRIPLET LOSS

FACENET: A UNIFIED EMBEDDING FOR FACE RECOGNITION AND CLUSTERING

- An image x is squeezed into a feature space \mathbb{R}^d , such that
 - the **squared distance** between all faces of the **same identity** - independently from imaging conditions - **is small**,
 - the **squared distance** between a pair of face images from **different identities** is **large**.
- An image x_a (**anchor**) of a specific person is closer to all other images x_p (**positive**) of the same person than it is to any image x_n (**negative**) of any other person.



FACENET: A UNIFIED EMBEDDING FOR FACE RECOGNITION AND CLUSTERING

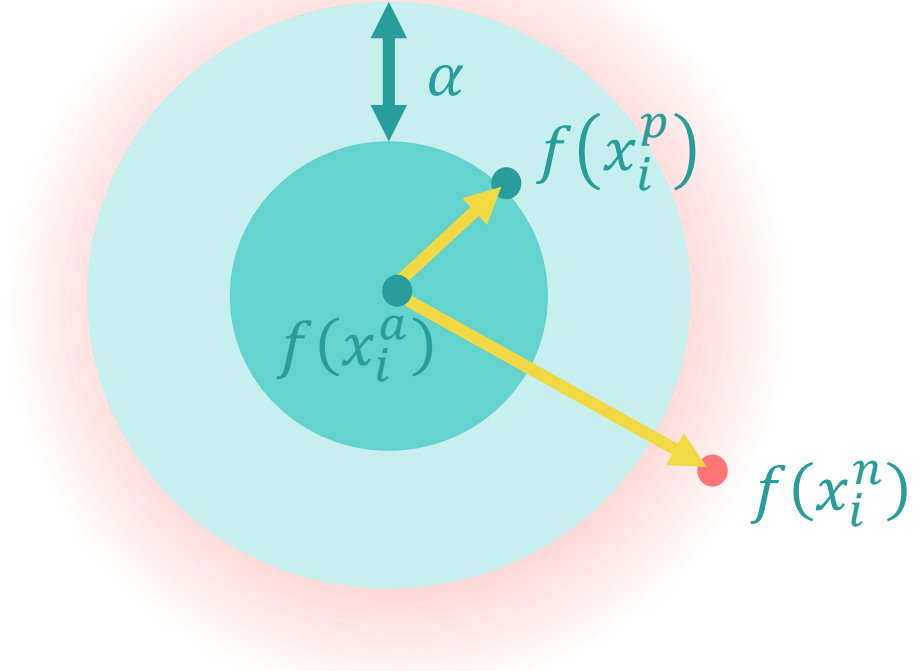
- Given a set of all possible triplets T ,

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

- TRIPLET LOSS:**

$$\sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+$$

- α is a margin that is enforced between positive and negative pairs.



FACENET: A UNIFIED EMBEDDING FOR FACE RECOGNITION AND CLUSTERING

- It is crucial to select hard triplets, that are active and can therefore contribute to improving the model.
- Sample mining:
 - **Offline:** at the beginning of each epoch compute all the embeddings on the training set, and then only select hard or semi-hard triplets. **Not efficient.**
 - **Online:** compute useful triplets on the fly, for each mini-batch of inputs



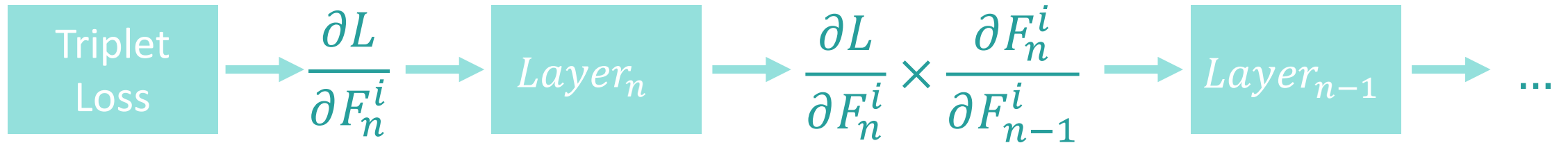
TRIPLER LOSS TRAINING ALGORITHM

1. Given a triplet $\{x_i^a, x_i^p, x_i^n\}_i$, compute the Forward pass for each element $\{f(x_i^a), f(x_i^p), f(x_i^n)\}_i$, (same net, same weights).
2. Compute the score with the Triplet loss.
3. Start computing the gradient of the Triplet loss with respect to the three outputs

$$L' = \begin{cases} \frac{\partial L}{\partial f(x_i^a)} = 2(f(x_i^n) - f(x_i^p)) & \text{if } \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha > 0 \\ \frac{\partial L}{\partial f(x_i^p)} = 2(f(x_i^p) - f(x_i^a)) & \text{if } \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha > 0 \\ \frac{\partial L}{\partial f(x_i^n)} = 2(f(x_i^a) - f(x_i^n)) & \text{if } \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha > 0 \end{cases}$$

TRIPLET LOSS TRAINING ALGORITHM

Given a generic weighted layer $\{F_n^i, W_n^i\}_n^i$ during the training step i



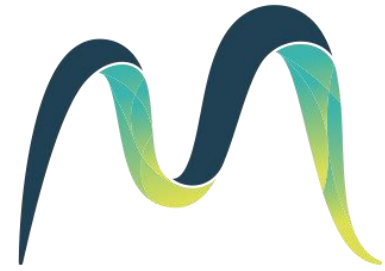
The weights are updated as:

$$W_n^i = W_n^i - \mu \left[\frac{\partial L}{\partial F_n^i} \times \frac{\partial F_n^i}{\partial W_n^i} \right]$$
$$= W_n^i - \mu \left[\frac{\partial L}{\partial F(X_a)_n^i} \times \frac{\partial F(X_a)_n^i}{\partial W_n^i} + \frac{\partial L}{\partial F(X_p)_n^i} \times \frac{\partial F(X_p)_n^i}{\partial W_n^i} + \frac{\partial L}{\partial F(X_n)_n^i} \times \frac{\partial F(X_n)_n^i}{\partial W_n^i} \right]$$

PART II

Python code

https://github.com/n-gregori/Deeplearning_lecture_SiameseNetworks



metaliquid

Thank you!