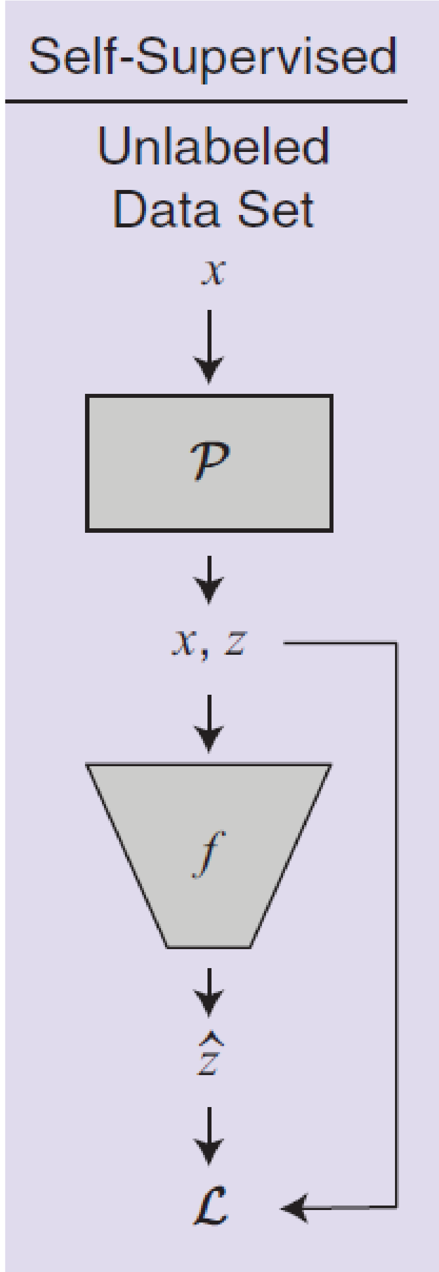
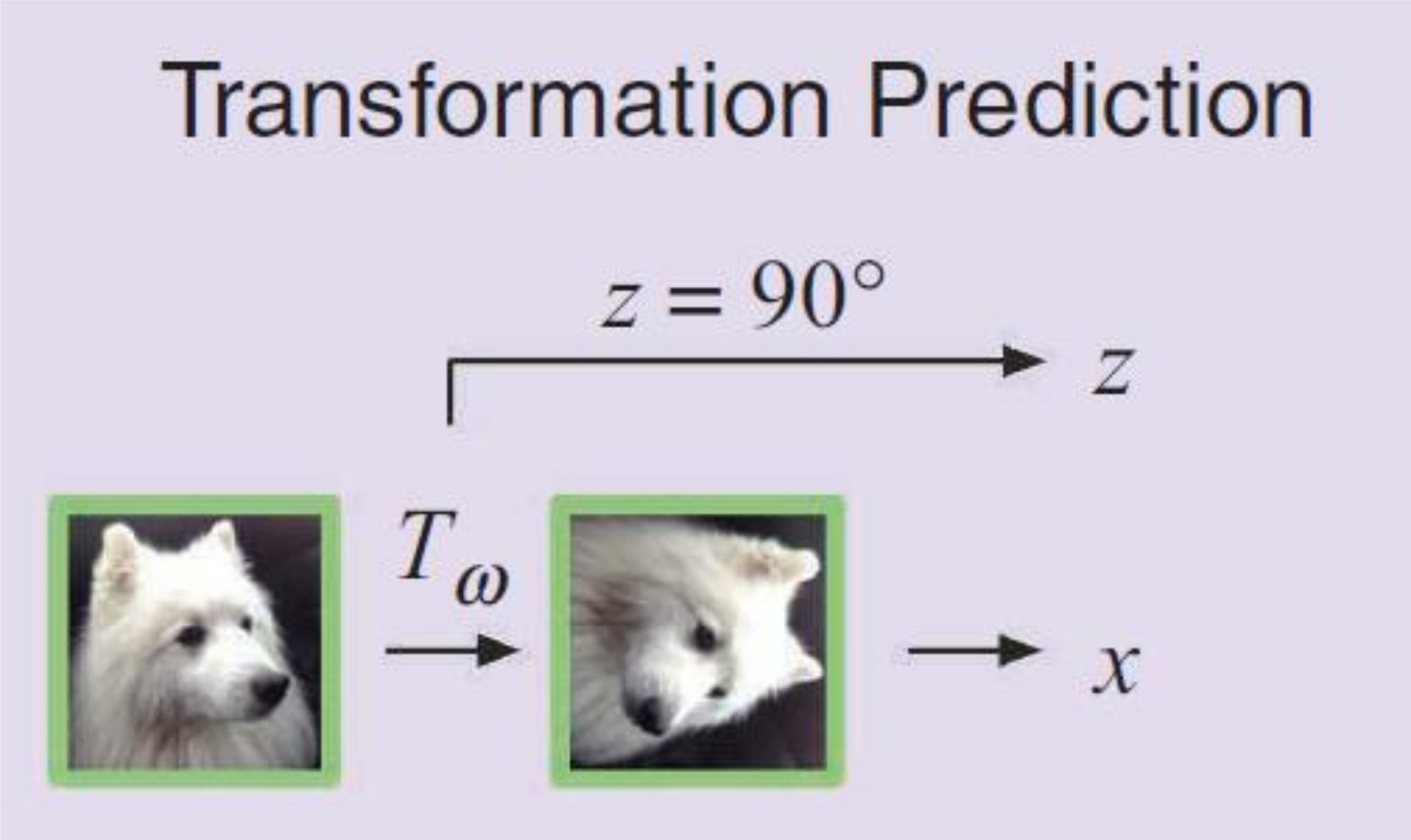
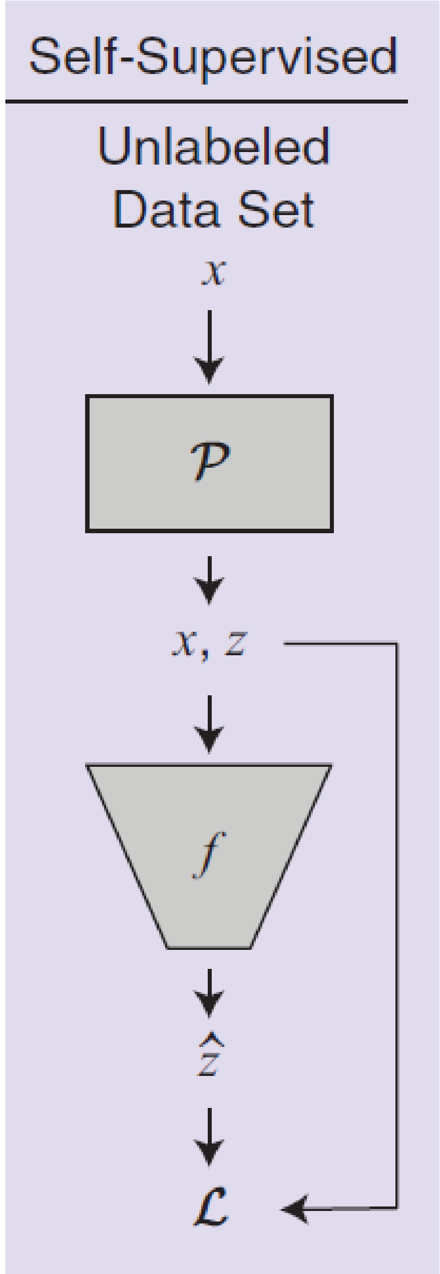
Pretext tasks
3. SimCLR

1 2	SELF-PREDICTION	INNATE RELATIONSHIP (Context-based)	1. ROTATION 2. RELATIVE POSITION	IMAGE
3	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	1. Instance Discrimination 2. SimCLR [Contrastive Loss] 3. Theory – Guarantees / Bounds	IMAGE
4	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	Contrastive Predictive Coding (CPC), [NCE, InfoNCE Loss]	AUDIO/ SPEECH
5	SELF-PREDICTION	GENERATIVE (VAE)	1. AE – Variational Bayes 2. VQ-VAE + AR	IMAGE AUDIO/ SPEECH
6	SELF-PREDICTION	GENERATIVE (AR)	1. AR-LM – GPT 2. Masked-LM – BERT	LANGUAGE
7	SELF-PREDICTION	MASKED-GEN (Masked LM for ASR)	1. Wav2Vec / 2.0 2. HuBERT	AUDIO/ SPEECH


Self-Supervised
Unlabeled
Data Set
x
$\mathcal{P}$
x, z
f
$\hat{z}$
$\mathcal{L}$


Transformation Prediction
$z = 90°$
z
T_ω
x



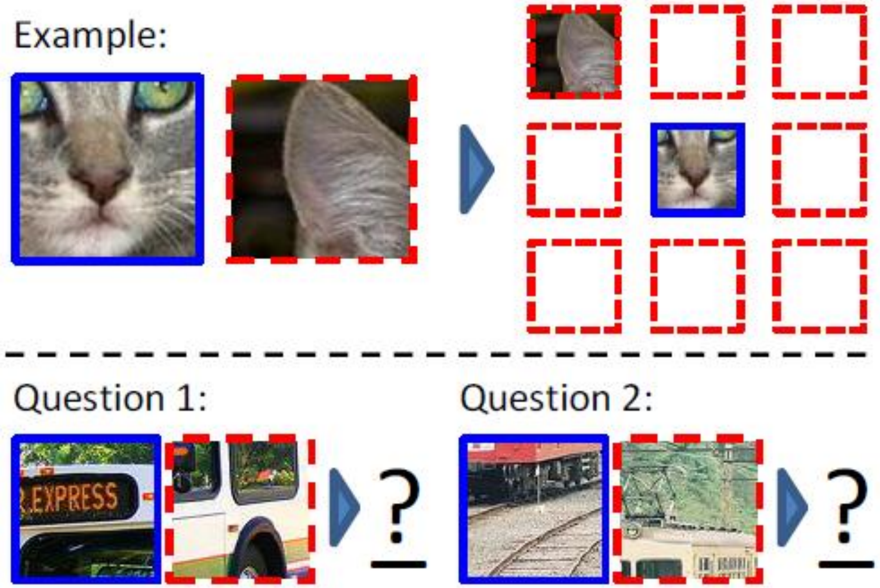
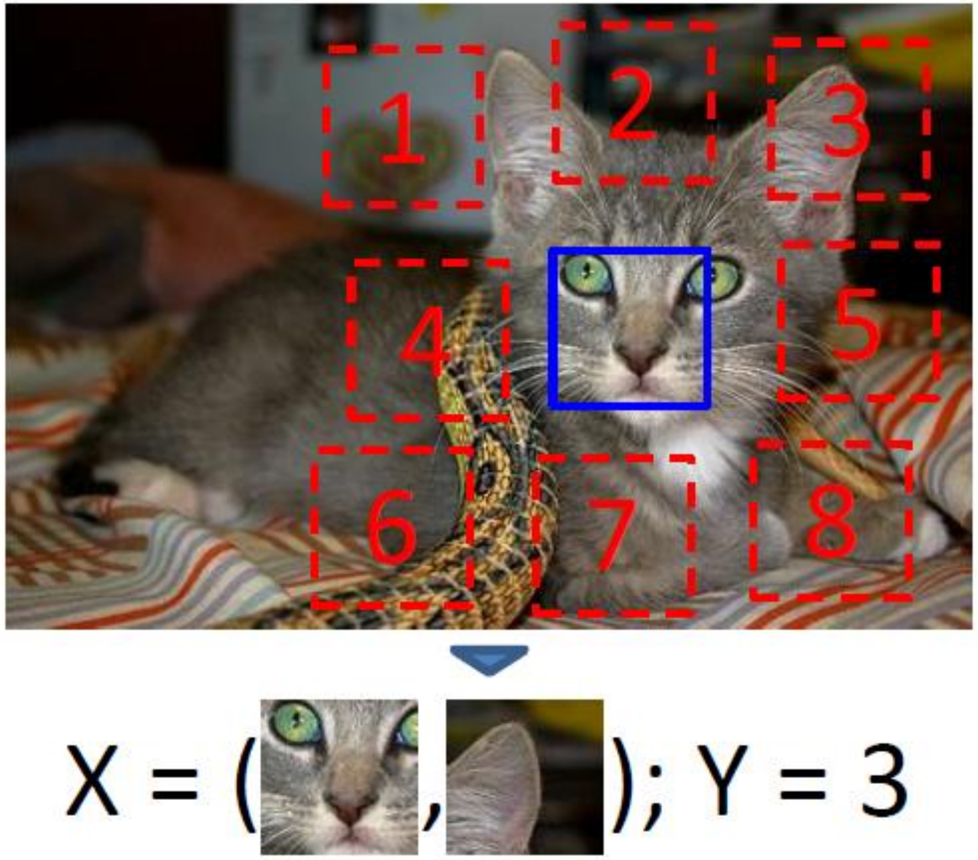


Figure 1. Our task for learning patch representations involves randomly sampling a patch (blue) and then one of eight possible neighbors (red). Can you guess the spatial configuration for the two pairs of patches? Note that the task is much easier once you have recognized the object!

Answer key: Q1: Bottom right Q2: Top center



1 2	SELF-PREDICTION	INNATE RELATIONSHIP (Context-based)	1. ROTATION 2. RELATIVE POSITION	IMAGE
3	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	1. Instance Discrimination 2. SimCLR [Contrastive Loss] 3. Theory – Guarantees / Bounds	IMAGE
4	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	Contrastive Predictive Coding (CPC), [NCE, InfoNCE Loss]	AUDIO/ SPEECH
5	SELF-PREDICTION	GENERATIVE (VAE)	1. AE – Variational Bayes 2. VQ-VAE + AR	IMAGE AUDIO/ SPEECH
6	SELF-PREDICTION	GENERATIVE (AR)	1. AR-LM – GPT 2. Masked-LM – BERT	LANGUAGE
7	SELF-PREDICTION	MASKED-GEN (Masked LM for ASR)	1. Wav2Vec / 2.0 2. HuBERT	AUDIO/ SPEECH

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen [1] **Simon Kornblith** [1] **Mohammad Norouzi** [1] **Geoffrey Hinton** [1]

[1]Google Research, Brain Team. Correspondence to: Ting Chen <iamtingchen@google.com>.

Proceedings of the 37^{th} *International Conference on Machine Learning*, Vienna, Austria, PMLR 119, 2020.

[1]Code available at https://github.com/google-research/simclr.

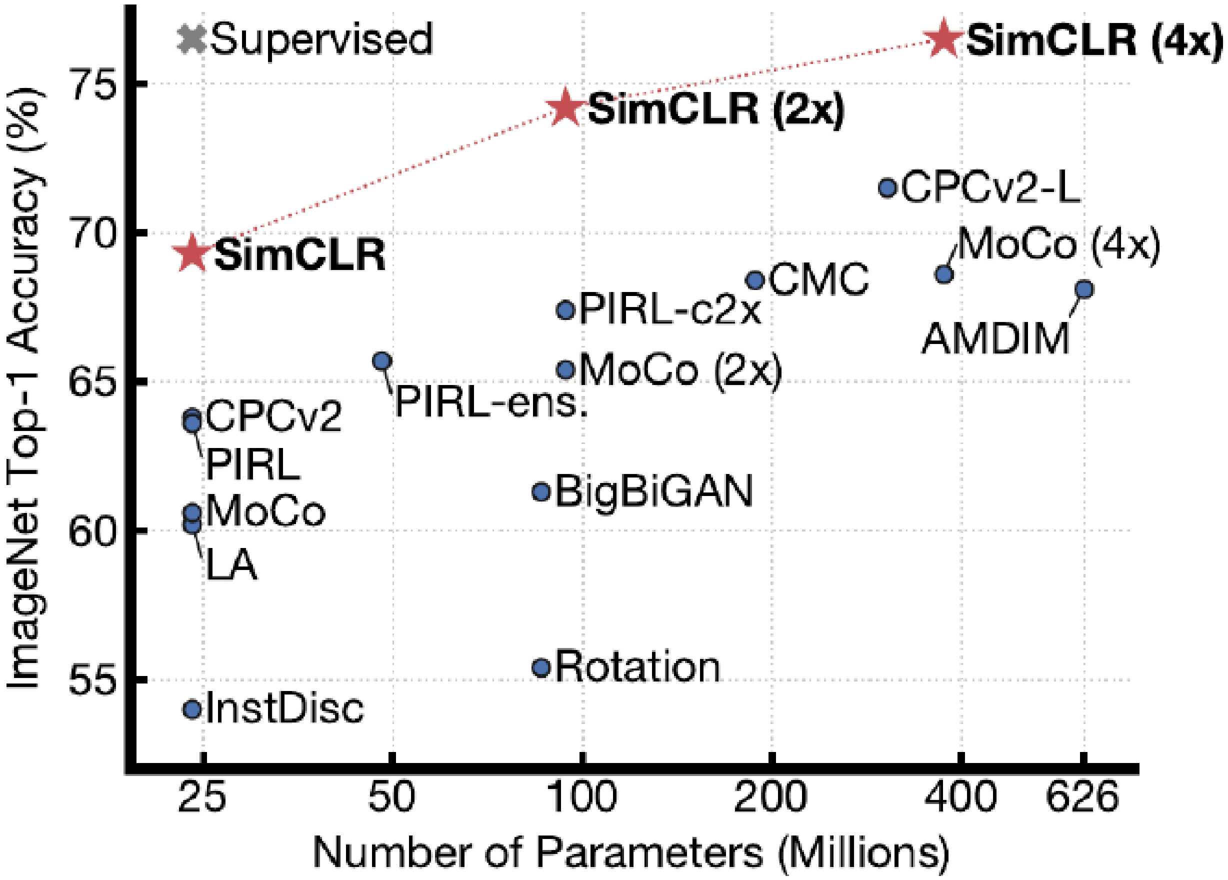


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

METRIC LEARNING:
SIAMESE NETWORKS

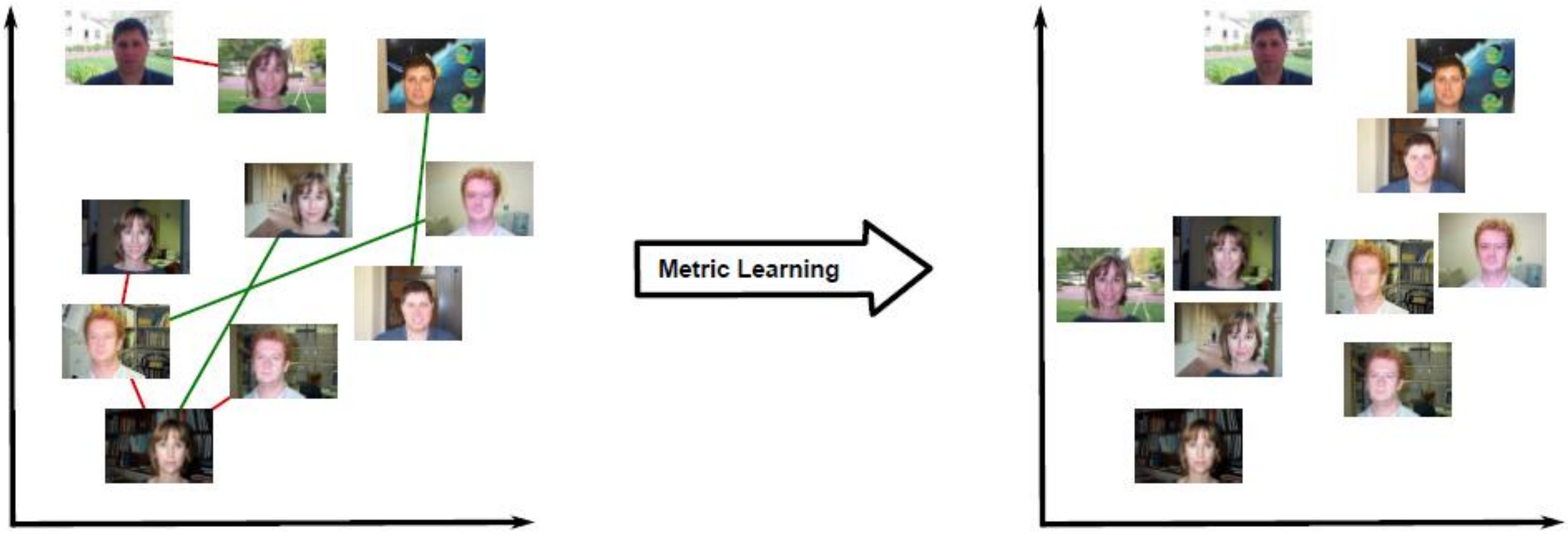
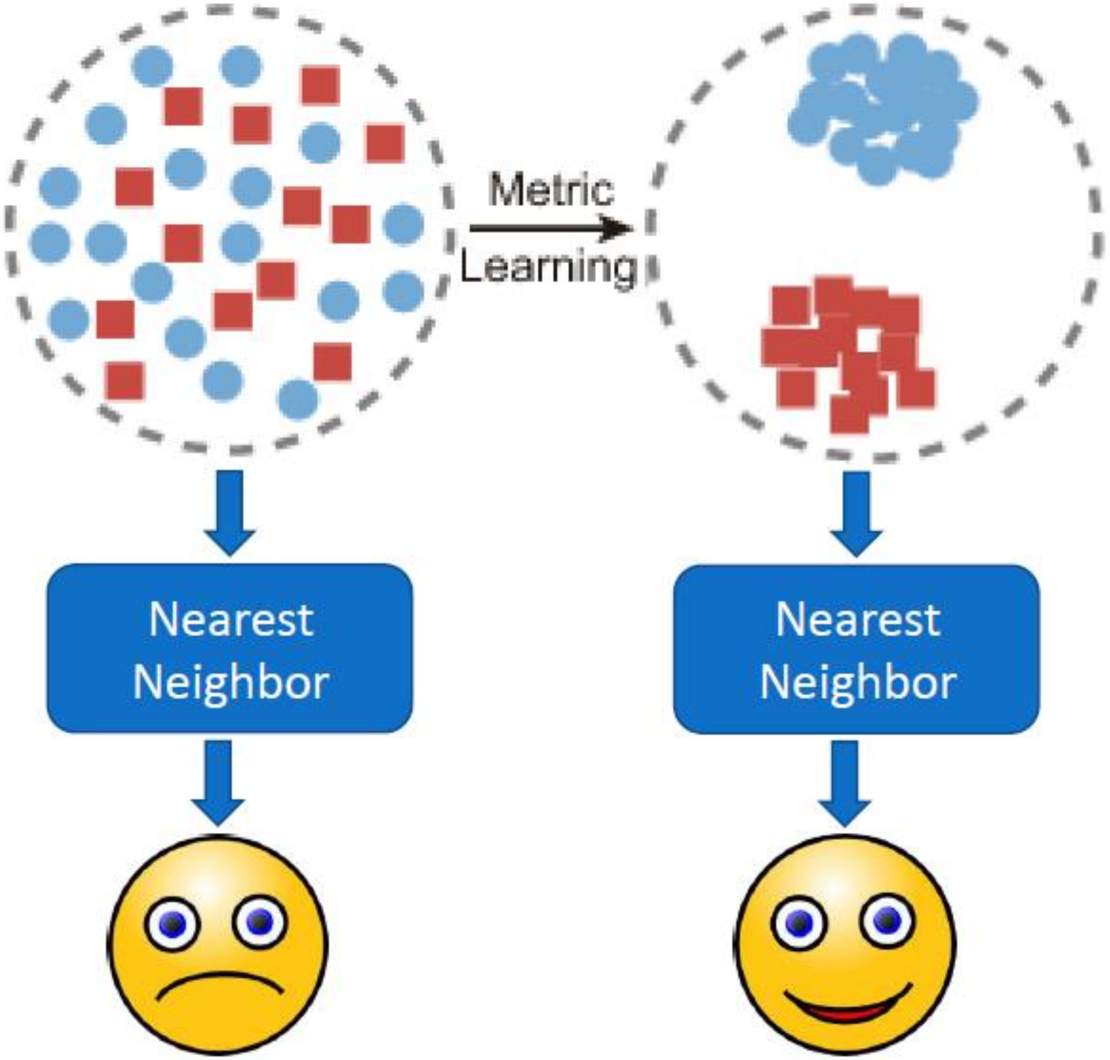


Figure 1: Illustration of metric learning applied to a face recognition task. For simplicity, images are represented as points in 2 dimensions. Pairwise constraints, shown in the left pane, are composed of images representing the same person (must-link, shown in green) or different persons (cannot-link, shown in red). We wish to adapt the metric so that there are fewer constraint violations (right pane). Images are taken from the Caltech Faces dataset.[8]

Metric Learning



Signature Verification using a "Siamese" Time Delay Neural Network

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger and Roopak Shah
AT&T Bell Laboratories
Holmdel, NJ 07733
jbromley@big.att.com

Abstract

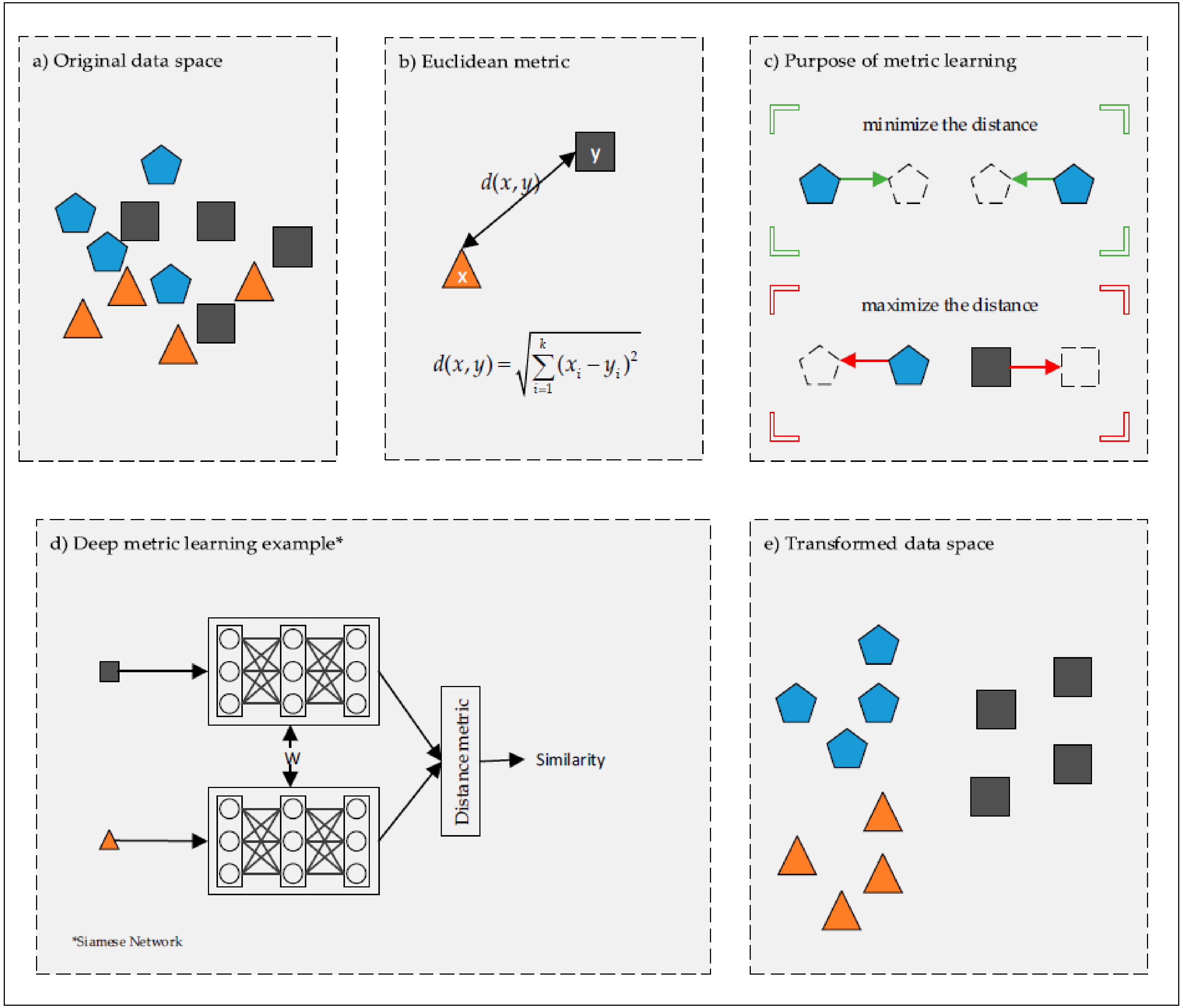
This paper describes an algorithm for verification of signatures written on a pen-input tablet. The algorithm is based on a novel, artificial neural network, called a "Siamese" neural network. This network consists of two identical sub-networks joined at their outputs. During training the two sub-networks extract features from two signatures, while the joining neuron measures the distance between the two feature vectors. Verification consists of comparing an extracted feature vector with a stored feature vector for the signer. Signatures closer to this stored representation than a chosen threshold are accepted, all other signatures are rejected as forgeries.

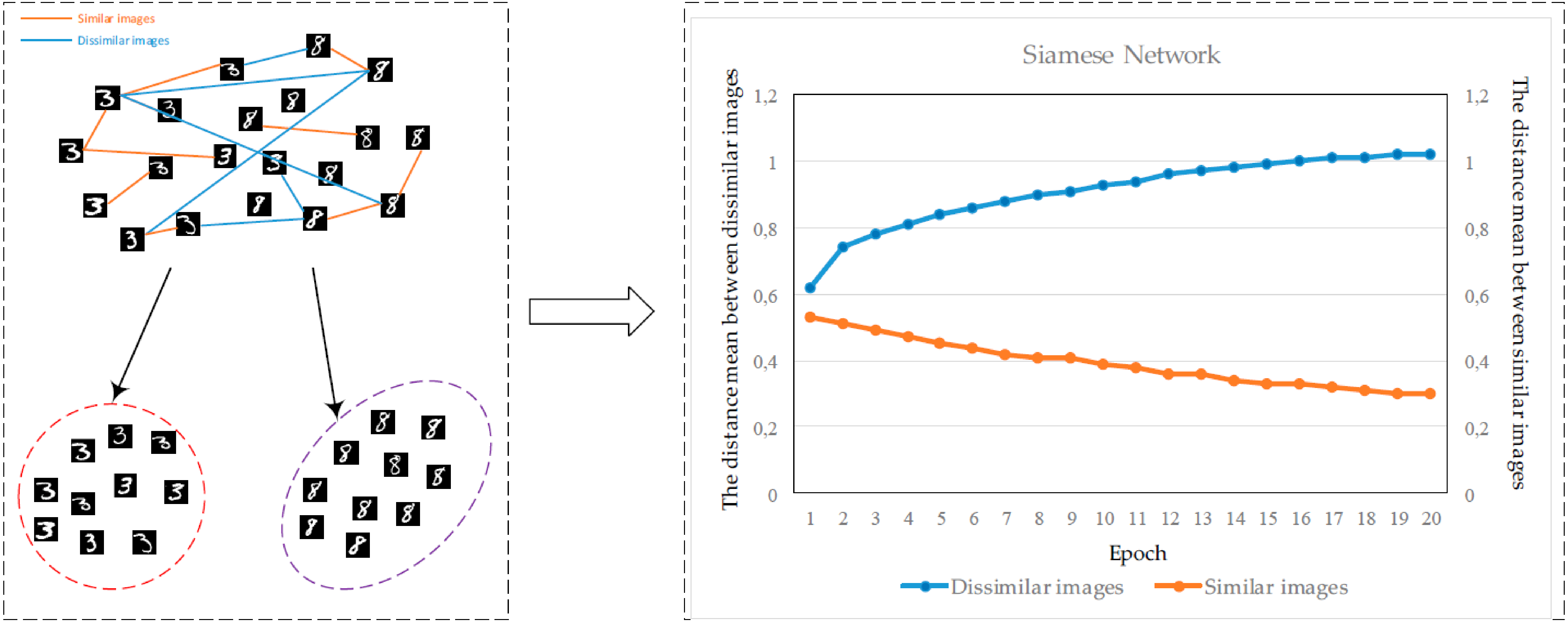
Bromley, Jane, Bentz, James W, Bottou, Léon, Guyon, Isabelle, LeCun, Yann, Moore, Cliff, Säckinger, Eduard, and Shah, Roopak. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7 (04):669–688, 1993.

[30] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn. (ICML) Deep Learn. Workshop*, vol. 2, 2015.

[98] M. Ye and Y. Guo, "Deep triplet ranking networks for one-shot recognition," *arXiv preprint arXiv:1804.07275*, 2018.

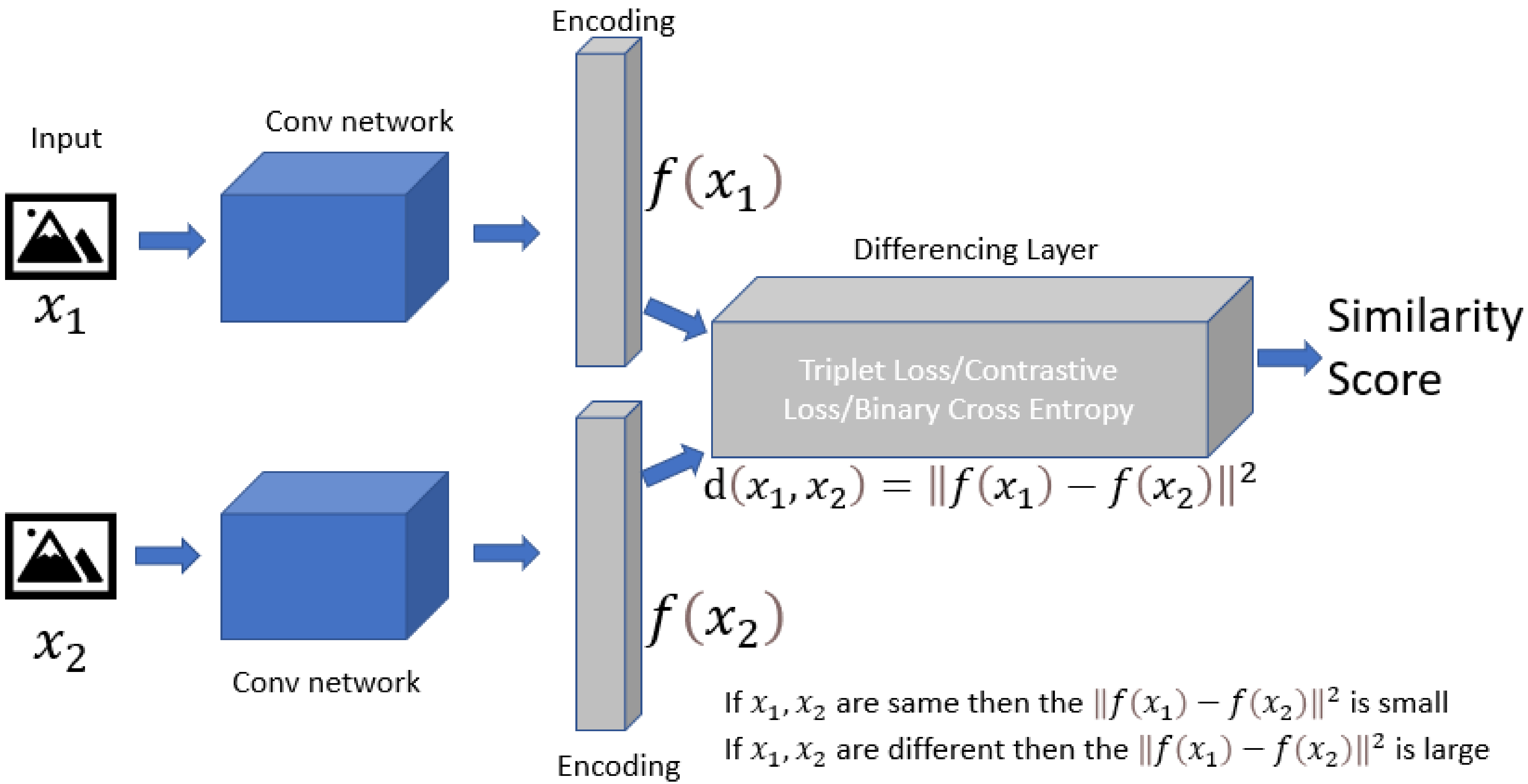
Deep Metric Learning





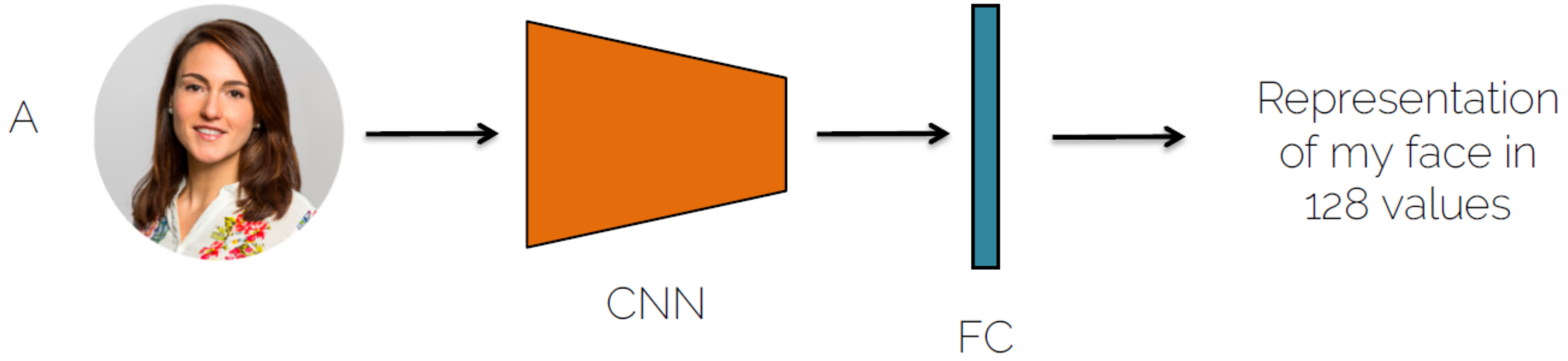
Similar images
Dissimilar images
Siamese Network
The distance mean between dissimilar images
The distance mean between similar images
1,2
1
0,8
0,6
0,4
0,2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Epoch
Dissimilar images
Similar images

- Siamese network takes two different inputs passed through two similar subnetworks with the same architecture, parameters, and weights.
- The two subnetworks are a mirror image of each other, just like the Siamese twins. Hence, any change to any subnetworks architecture, parameter, or weights is also applied to the other subnetwork.
- The two subnetwork outputs an encoding to calculate the difference between the two inputs.
- The Siamese network's objective is to classify if the two inputs are the same or different using the Similarity score. The Similarity score can be calculated using Binary cross-entropy, Contrastive function, or Triplet loss, which are techniques for the general distance metric learning approach.
- Siamese network is a one-shot classifier that uses discriminative features to generalize the unfamiliar categories from an unknown distribution.

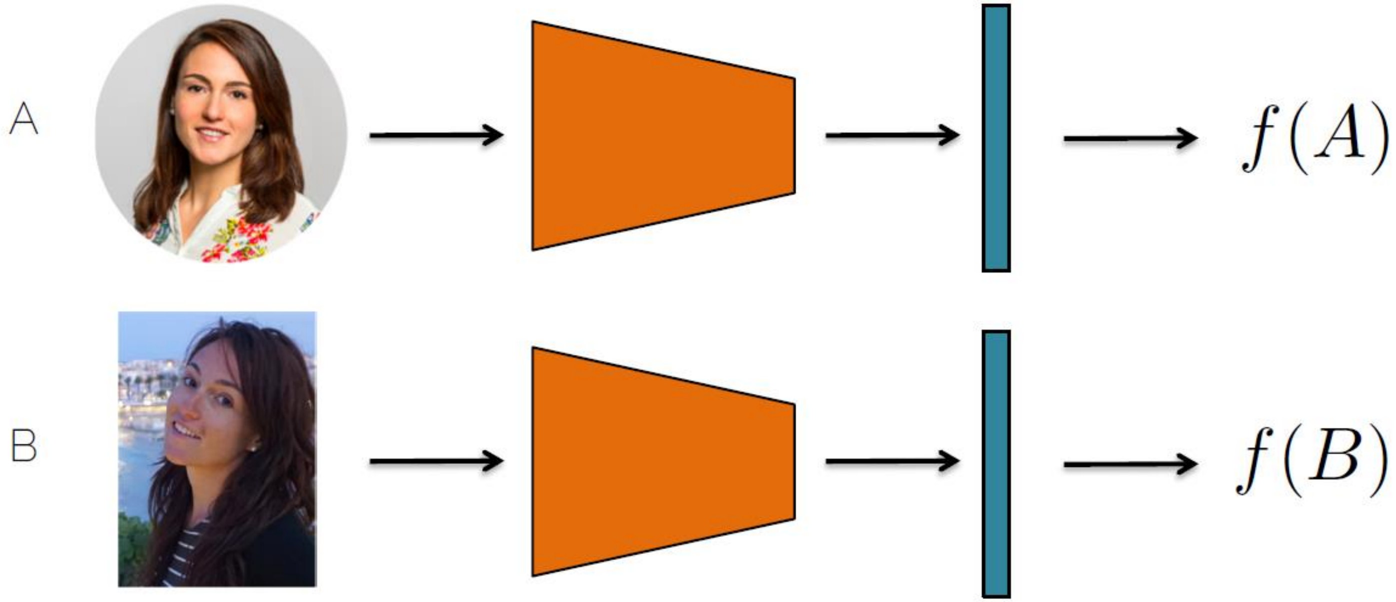

Encoding
Input
Conv network
$f(x_1)$
Differencing Layer
x_1
Similarity Score
Triplet Loss/Contrastive Loss/Binary Cross Entropy
$d(x_1, x_2) = \|f(x_1) - f(x_2)\|^2$
x_2
$f(x_2)$
Conv network
If x_1, x_2 are same then the $\|f(x_1) - f(x_2)\|^2$ is small
If x_1, x_2 are different then the $\|f(x_1) - f(x_2)\|^2$ is large
Encoding

Similarity learning

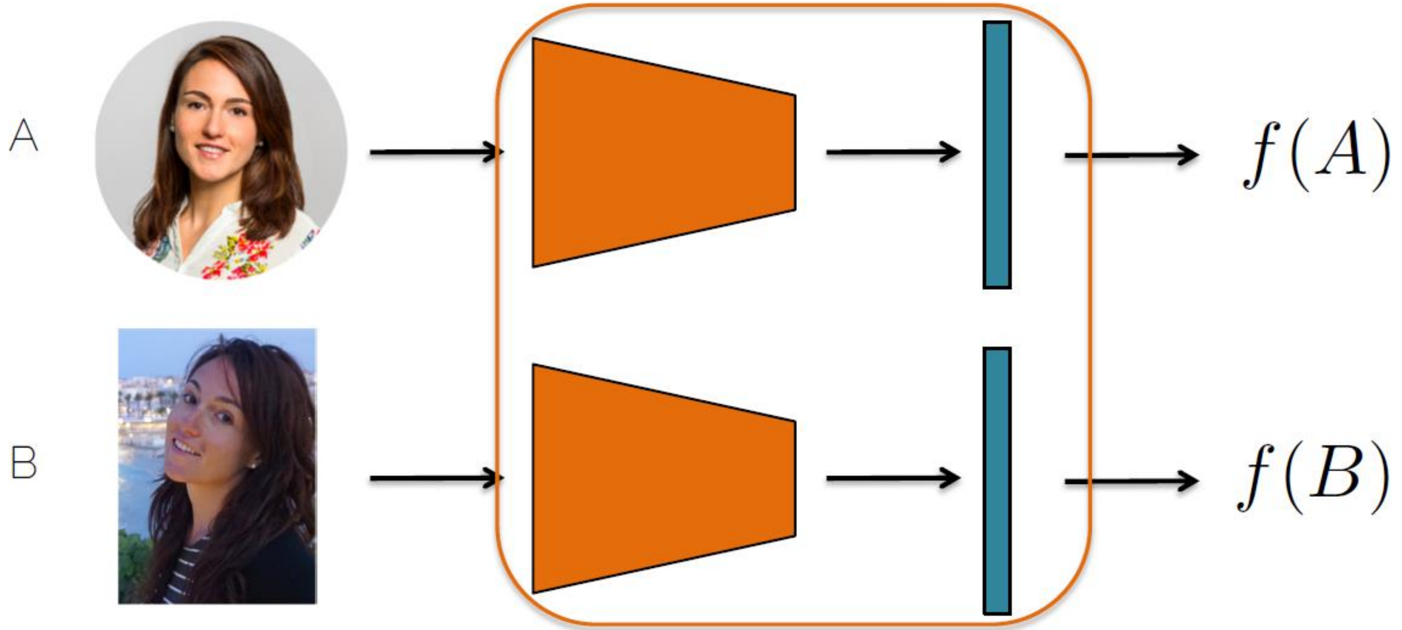
- How do we train a network to learn similarity?



- How do we train a network to learn similarity?



- Siamese network = shared weights

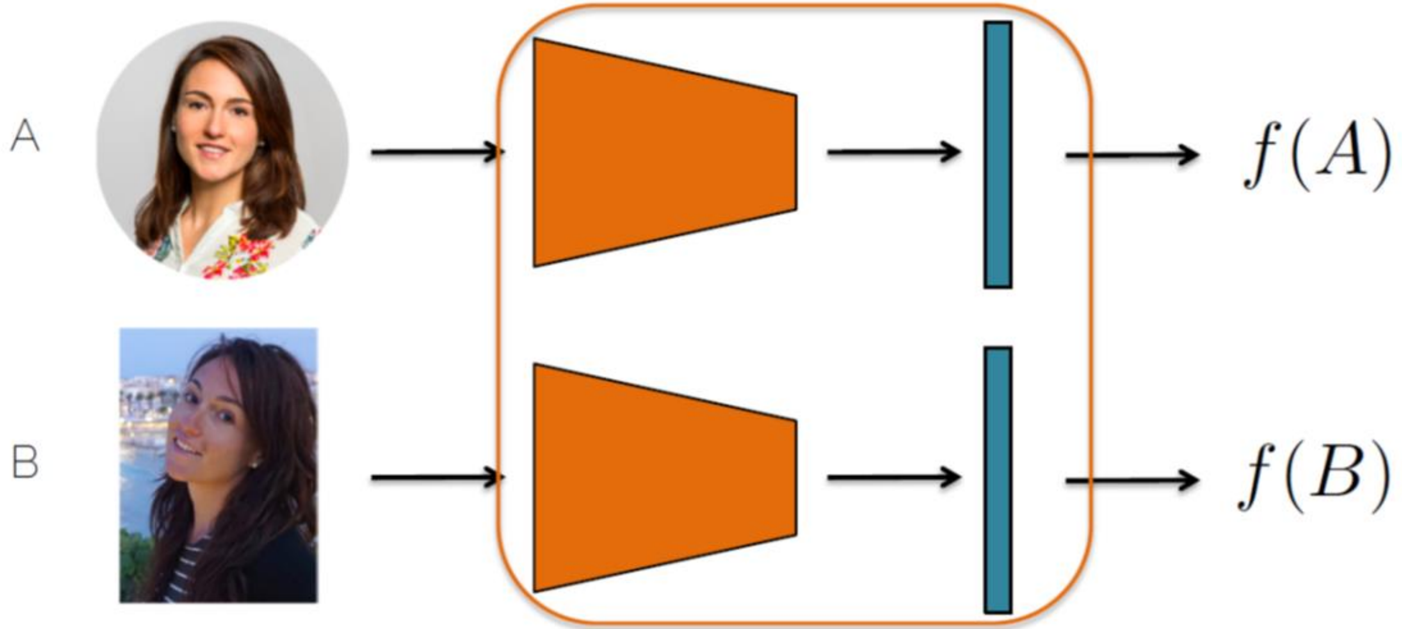


- Siamese network = shared weights

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

- To be done: compare the encodings

- Siamese network = shared weights



- 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

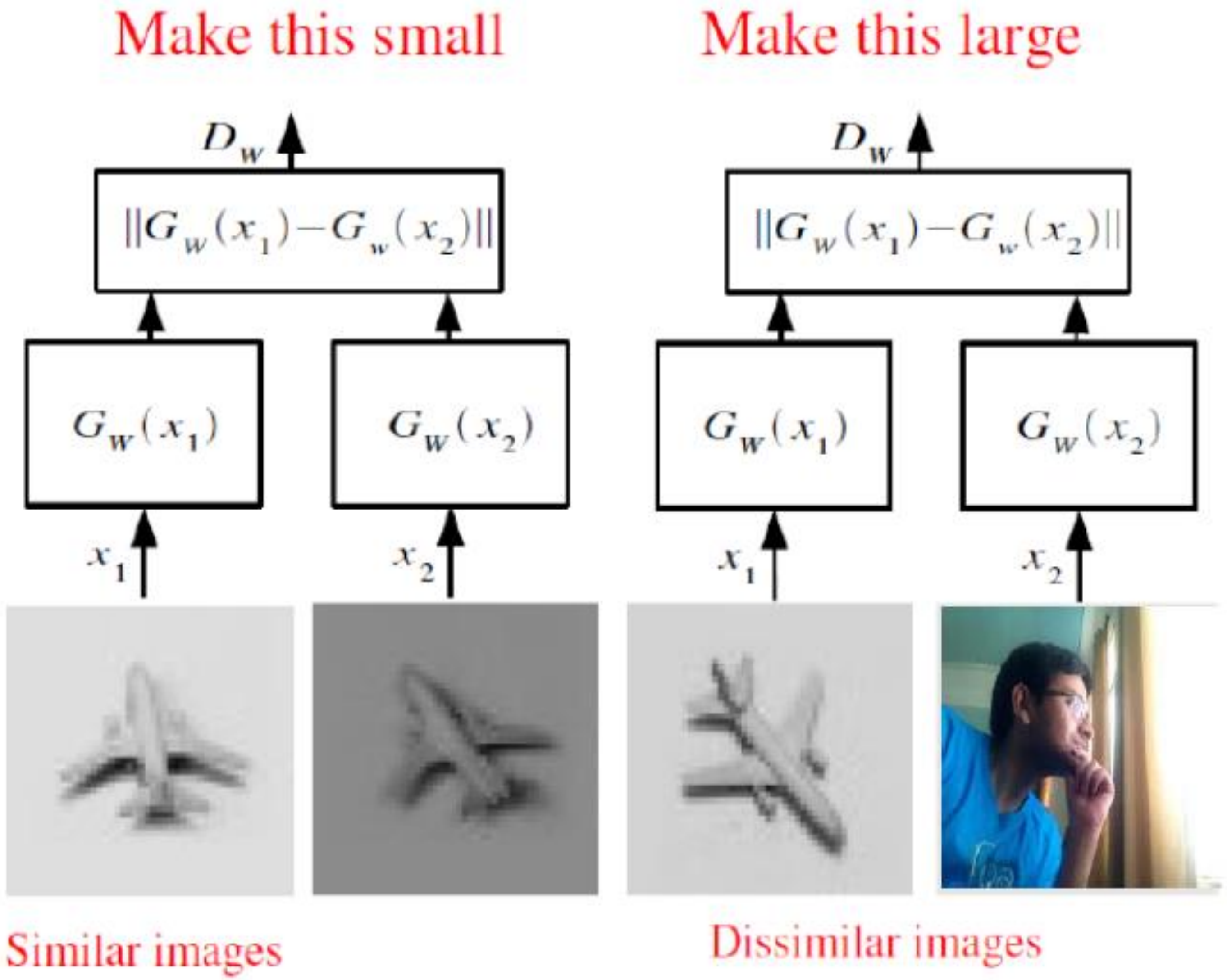
- Loss function for a positive pair:
  - If A and B depict the same person, $d(A, B)$ is small

$$\mathcal{L}(A, B) = ||f(A) - f(B)||^2$$

- Loss function for a negative pair:
  - If A and B depict a different person, $d(A, B)$ is large
  - Better use a Hinge loss:

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

If two elements are already far away, do not spend energy in pulling them even further away



The final loss is defined as :

L = ∑loss of positive pairs + ∑ loss of negative pairs

- Contrastive loss:

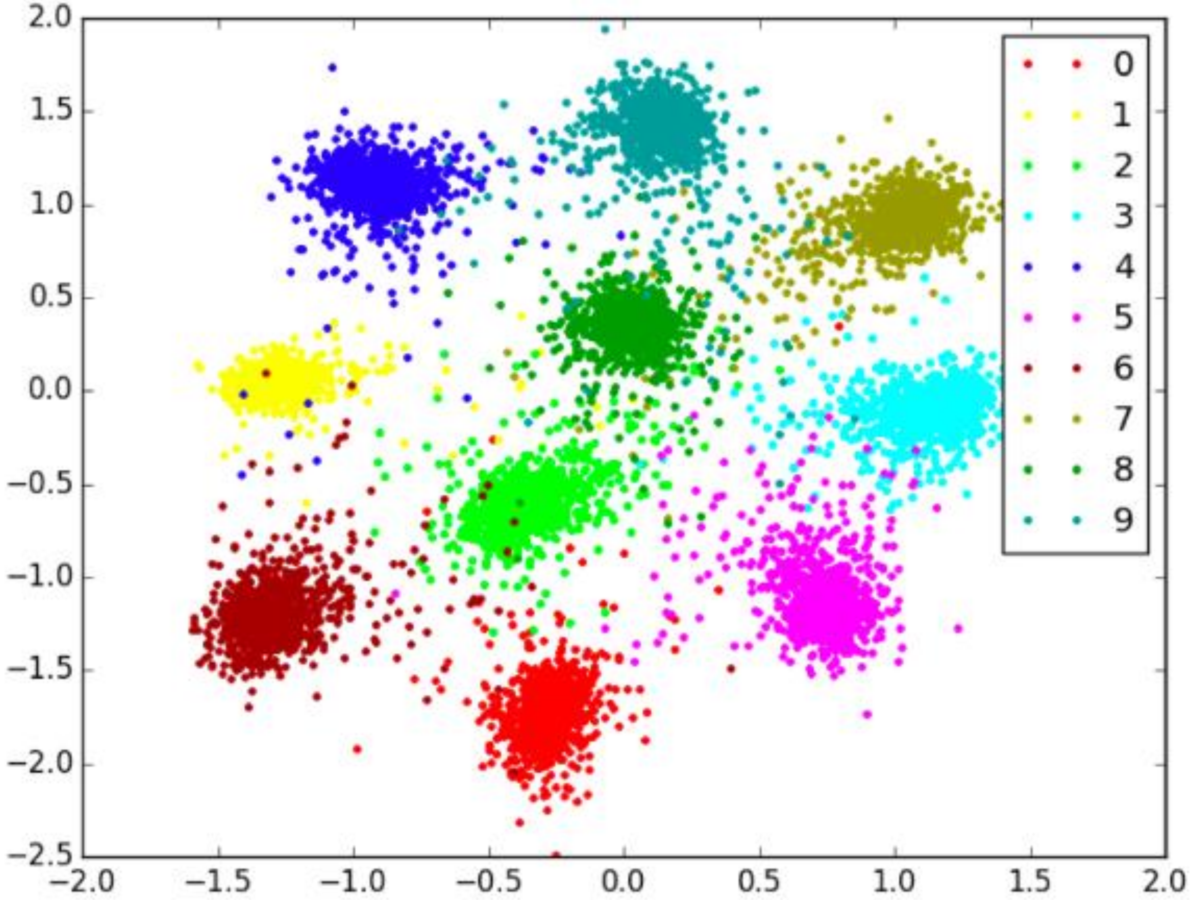
$$\mathcal{L}(A,B) = y^*||f(A)-f(B)||^2 + (1-y^*)max(0, m^2 - ||f(A)-f(B)||^2)$$

Positive pair, reduce the distance between the elements

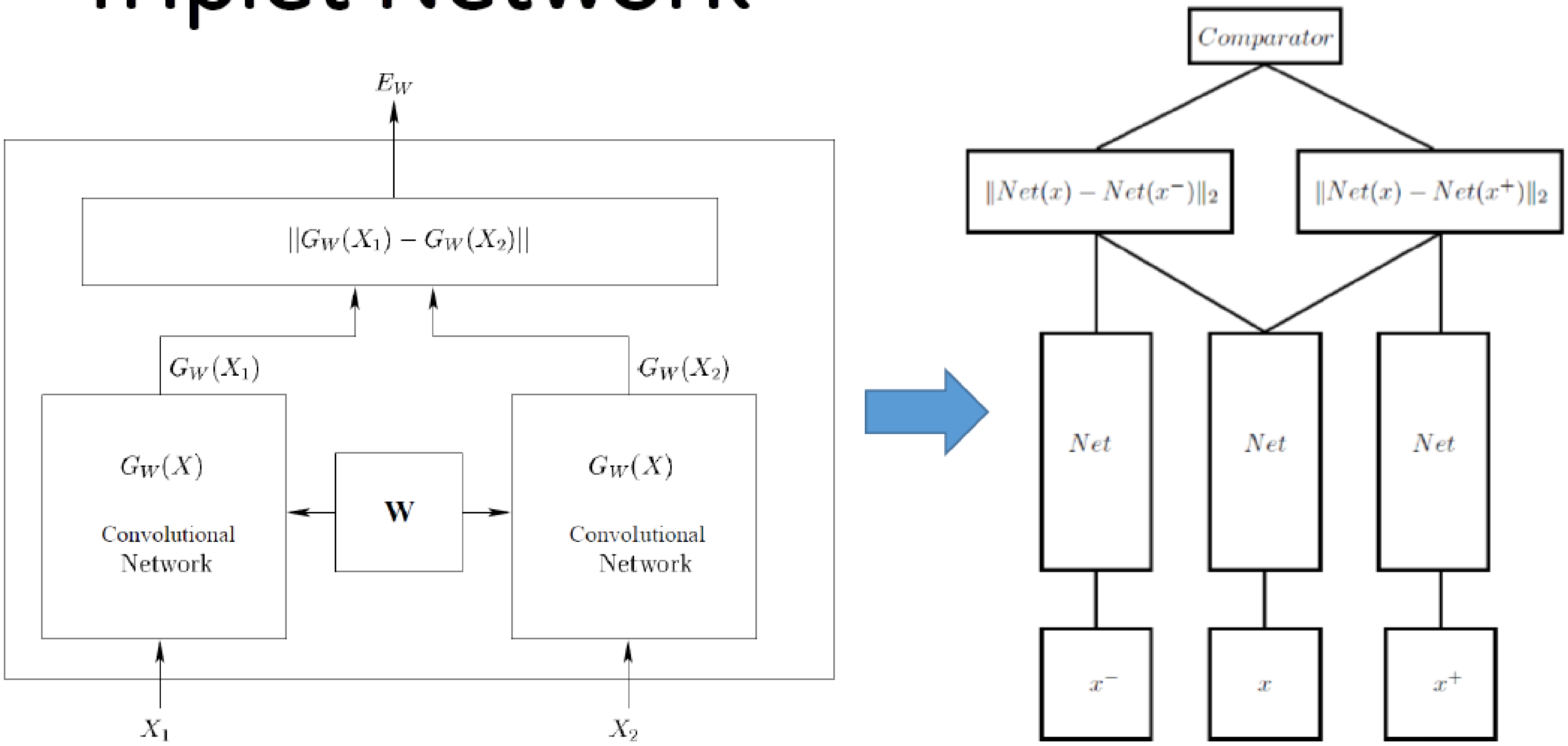
Negative pair, brings the elements further apart up to a margin

- 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

Siamese network on MNIST



Triplet Network



From Siamese to Triplet Network

Triplet loss

- Triplet loss allows us to learn a ranking

Anchor (A)

Positive (P)

Negative (N)

We want: $||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$

- 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

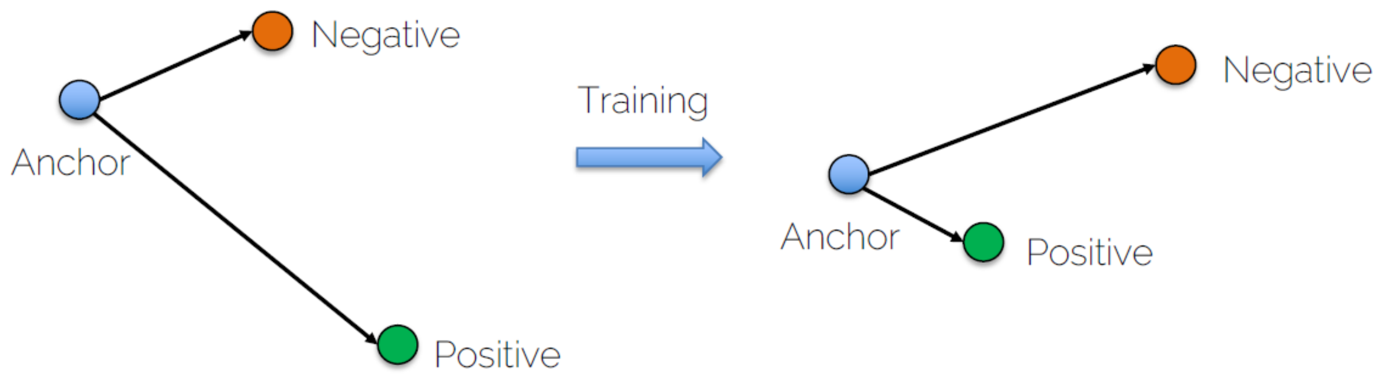
- Triplet loss allows us to learn a ranking

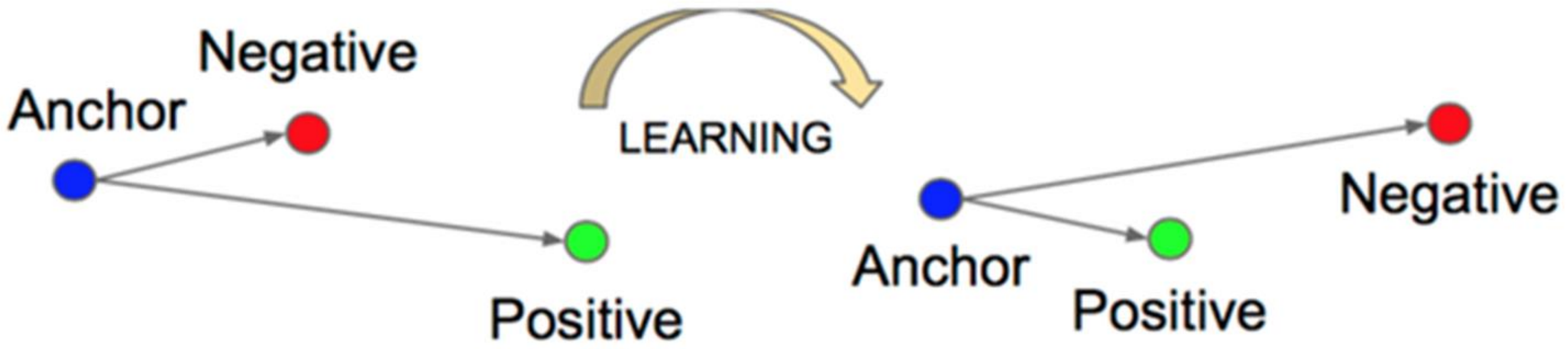
$$||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 < 0$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m < 0$$

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





Distance based loss function that operates on three inputs:

1. Anchor (a) is any arbitrary data point,
2. Positive (p) which is the same class as the anchor
3. Negative (n) which is a different class from the anchor

$$L=\max(d(a,p)-d(a,n)+margin,0)$$

We minimize this loss, which pushes ***d(a,p)*** to 0 and ***d(a,n)*** to be greater than ***d(a,p)+margin***. This means that, after the training, the positive examples will be closer to the anchor while the negative examples will be farther from it

Thank you !!