Занятие 10
Распознавание людей

Дмитрий Яшунин, к.ф.-м.н
IntelliVision

yashuninda@yandex.ru

Организационные моменты

Intelli-Vision was acquired by Nortek Security & Control

В прошлый раз: **Supervised Learning**

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, etc.

Cat

Classification

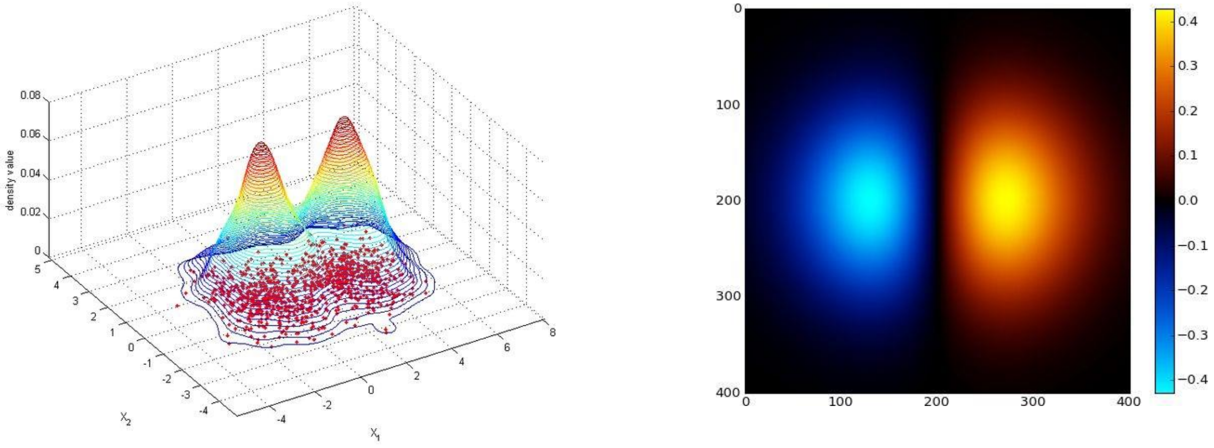
В прошлый раз: **Unsupervised Learning**

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, etc.

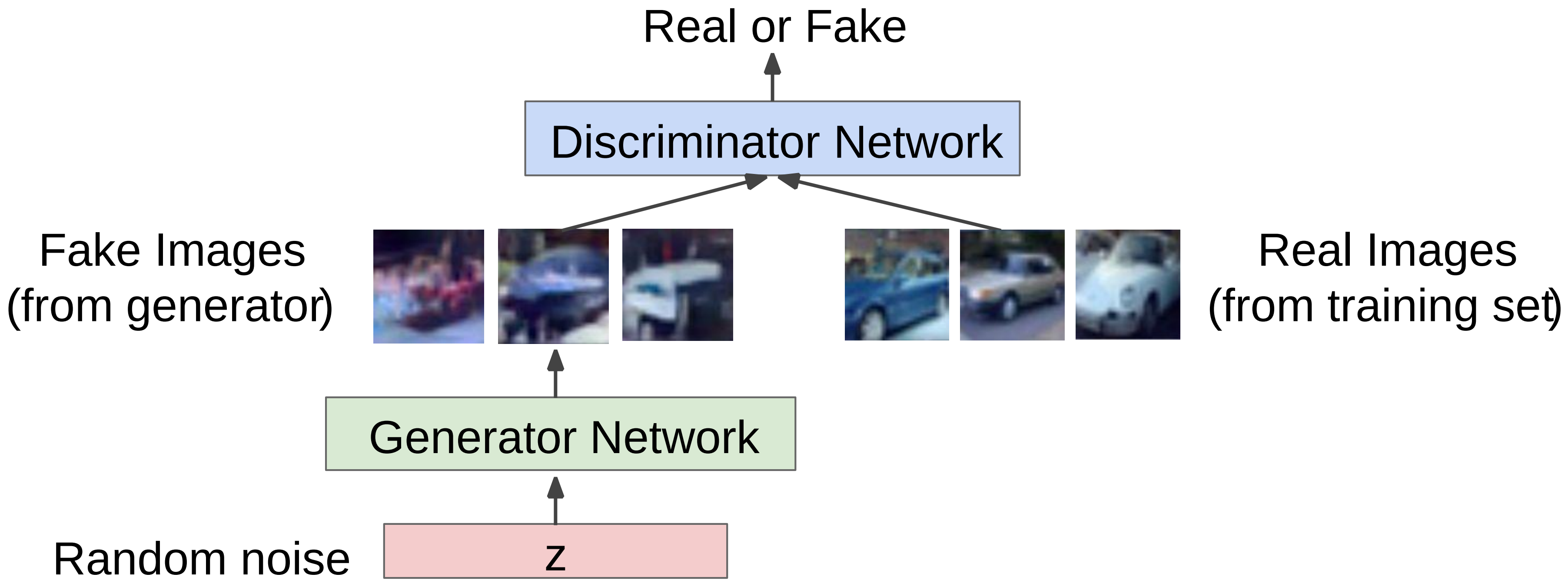
1-d density estimation



2-d density estimation

В прошлый раз: **GANs**

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



Сегодня:
Распознавание людей

Image similarity: **Image Retrieval**

Query Retrieval

Example retrieval results on Stanford Online Products

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

Image similarity: **Face recognition**

Example face images in MegaFace dataset

Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image similarity: **Person Reidentification**

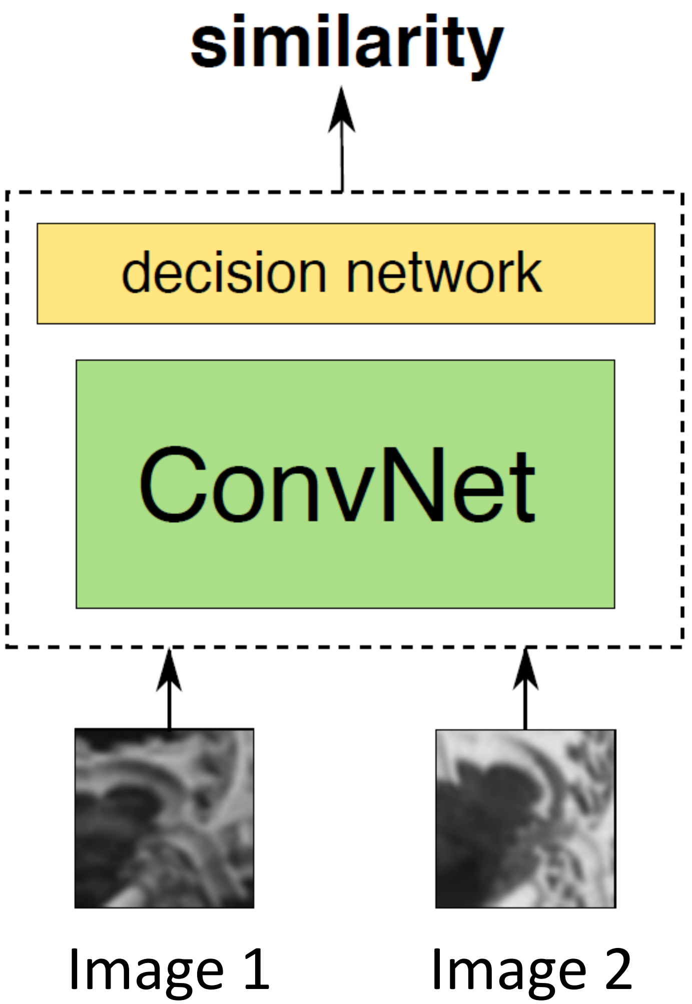
Xiao et al, "Joint Detection and Identification Feature Learning for Person Search", 2017

Image similarity

Image similarity
Image retrieval
Zero/one shot learning
Face recognition
Person reidentification (ReID)

The same task:
find “similar” images

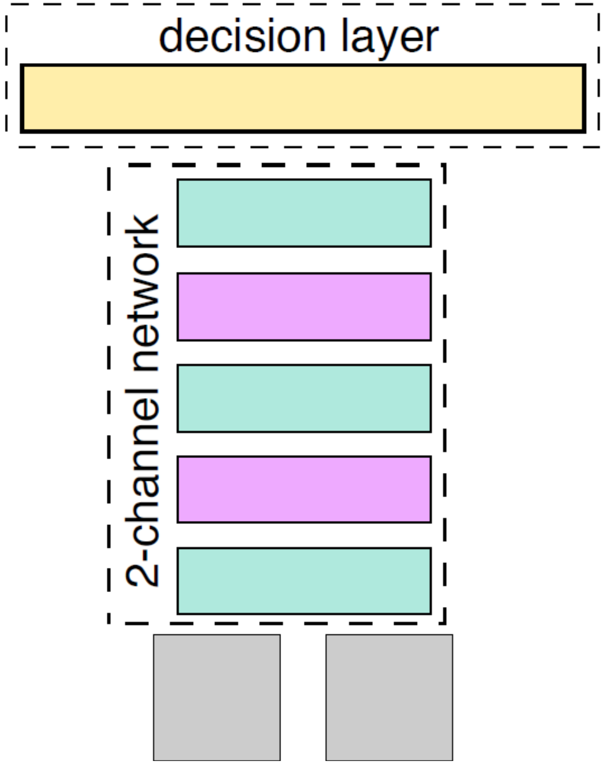
Image similarity: **Learning Similarity Function**

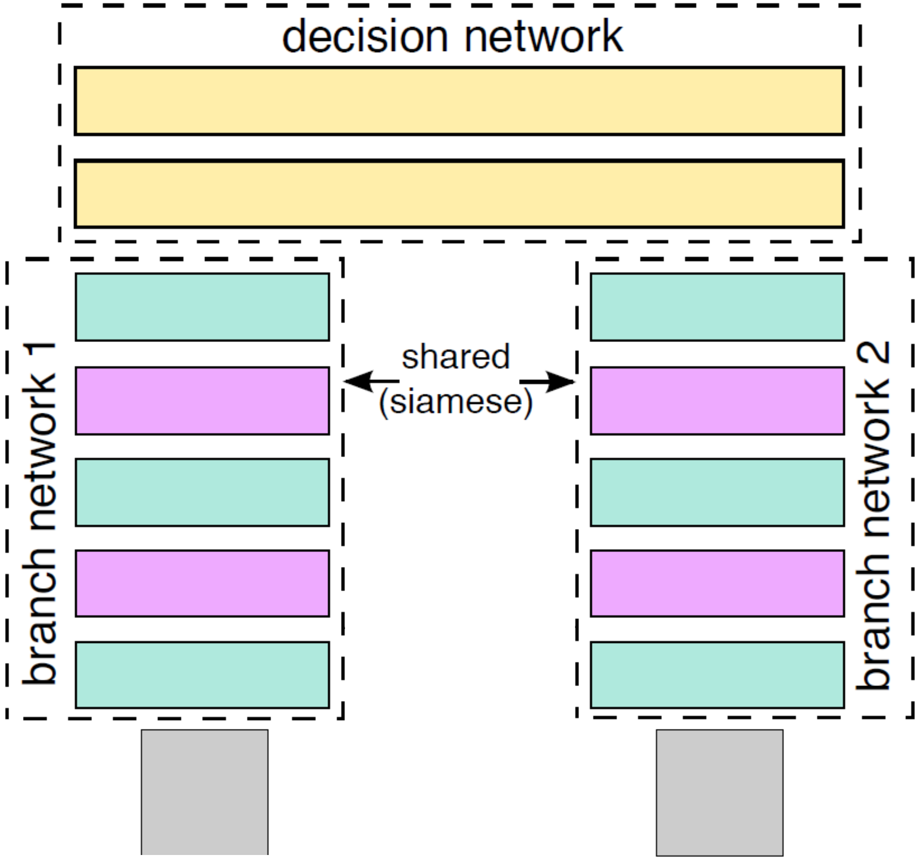


Zagoruyko and Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks", 2015

Image similarity: **Learning Similarity Function**

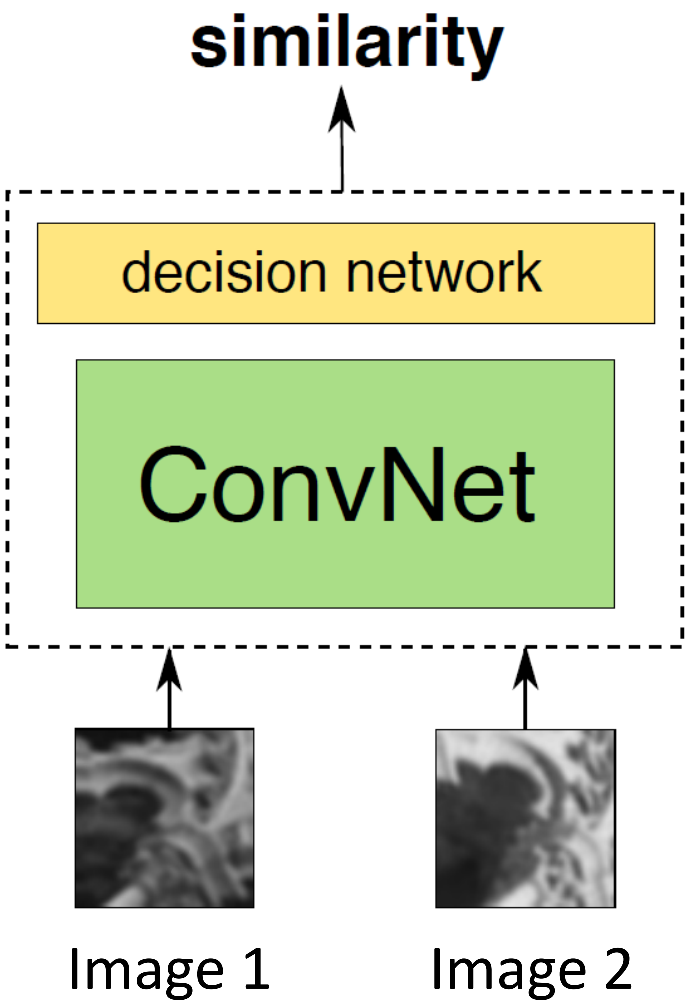
2-channel | Siamese network





Zagoruyko and Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks", 2015

Image similarity: **Learning Similarity Function**



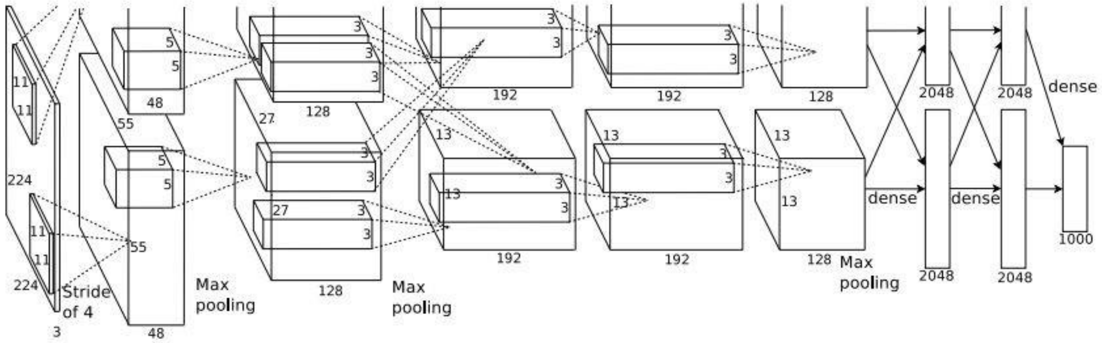
Very slow for many pairs

Zagoruyko and Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks", 2015

Image Similarity: **Embeddings**

Image

CNN



Embedding

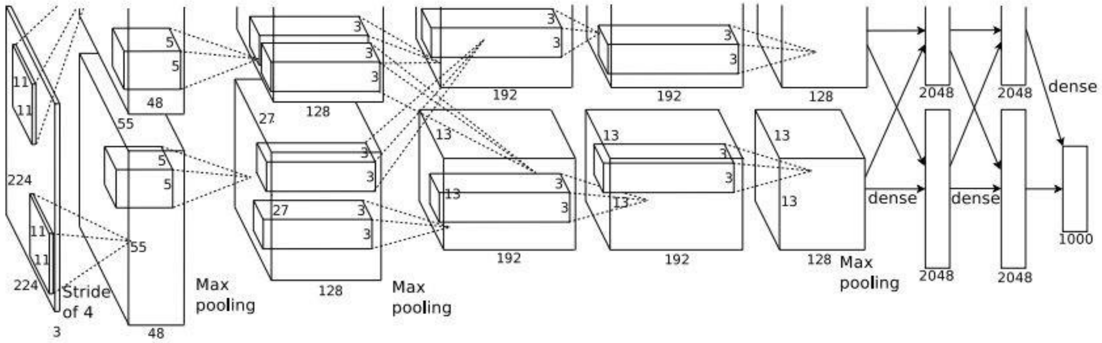
vector of fixed size

typical 2^n: 128, 256, 512

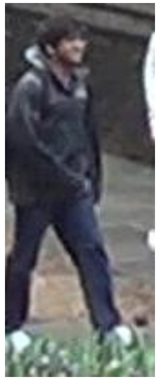
Image Similarity: **Embeddings**

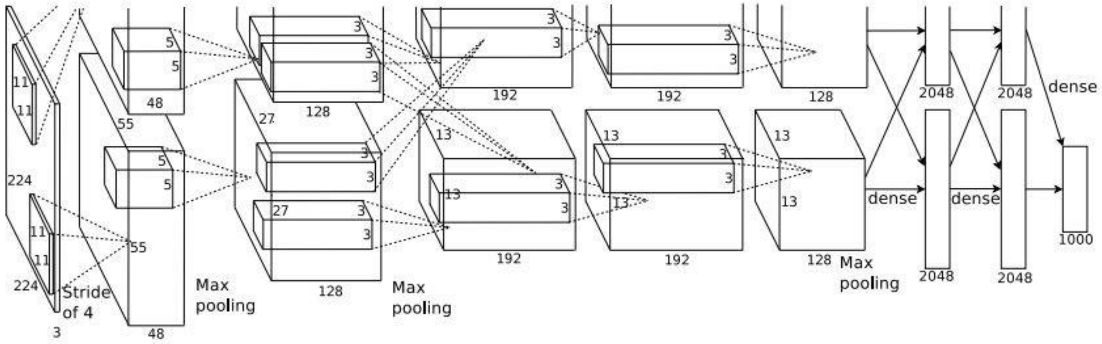
Image

CNN



Embedding 1



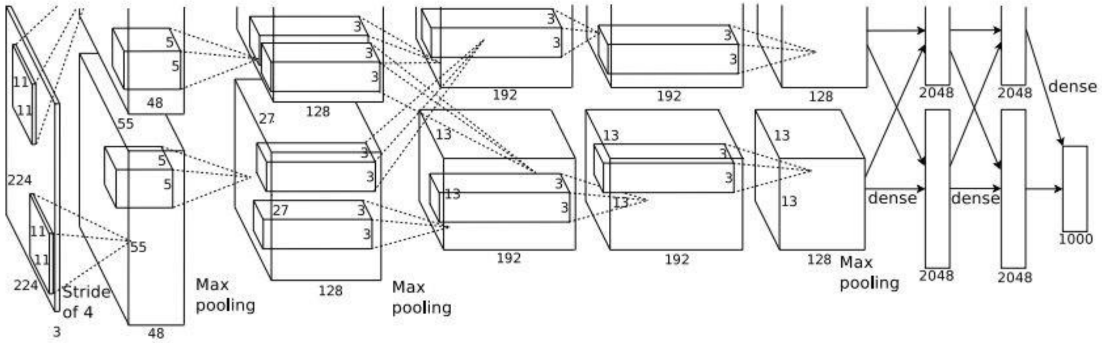


Embedding 2

Image Similarity: **Embeddings**

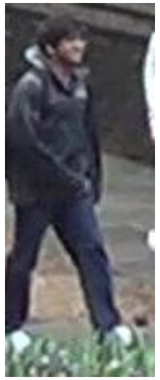
Image

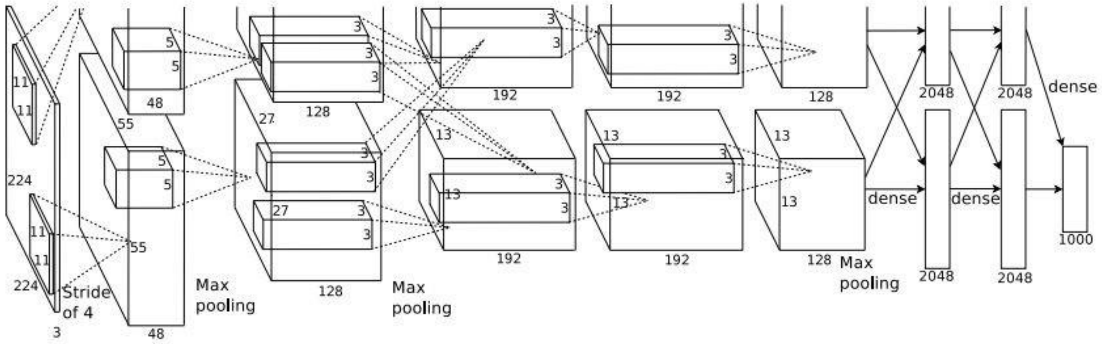
CNN



Embedding 1

compare l2 distance between embeddings





Embedding 2

Image Similarity: Loss functions: **Softmax**

Test image   L2 Nearest neighbors in feature space

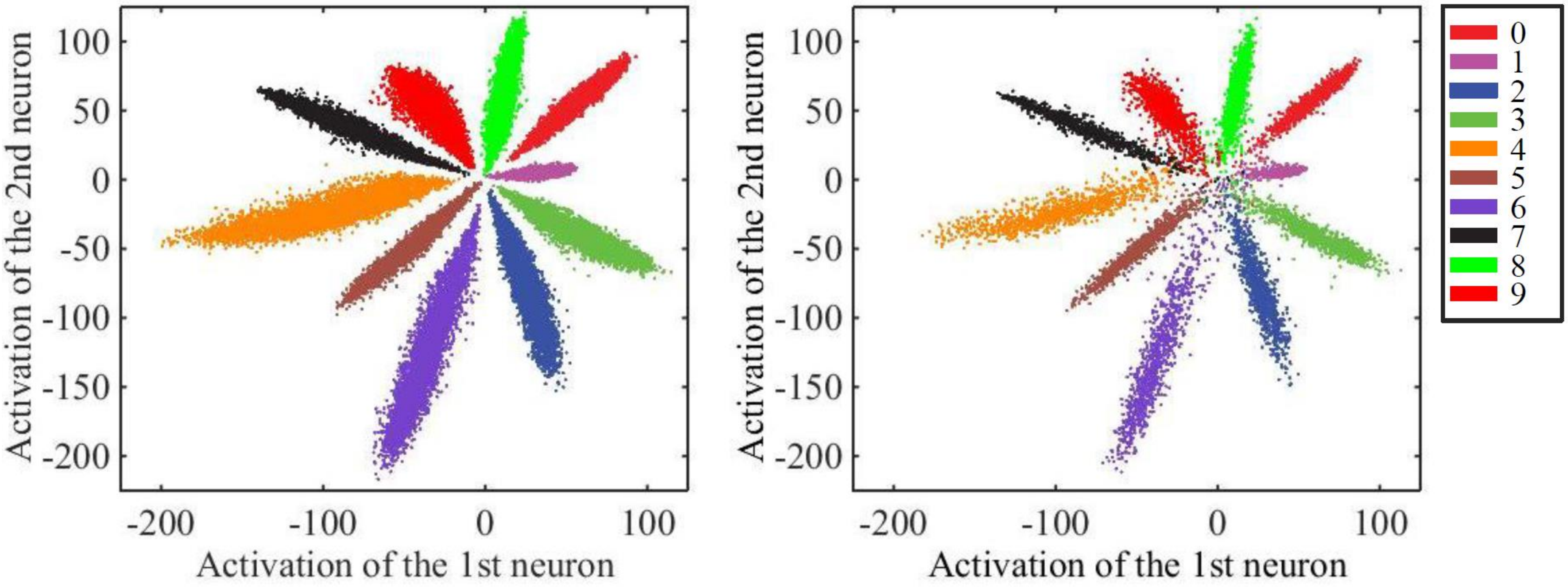
Solve classification task

Use output of last layer before softmax as embedding

Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012

Image Similarity: Loss functions: **Softmax**

MNIST handwritten digits 0-9



train

test

Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image Similarity: Loss functions: **Center Loss**

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}}$$

Softmax loss

Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image Similarity: Loss functions: **Center Loss**

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boxed{\boldsymbol{x}_i} + b_j}}$$

Softmax loss

x_i – output features (before softmax layer)

Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image Similarity: Loss functions: **Center Loss**

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}}$$

Softmax loss

x_i – output features (before softmax layer)

$$\mathcal{L}_C = \frac{1}{2} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boxed{\boldsymbol{c}_{y_i}}\|_2^2$$

Center loss: direct clustering of features around centers

c_{yi} – learnable centers for classes y_i

Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image Similarity: Loss functions: **Center Loss**

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}}$$

Softmax loss

x_i – output features (before softmax layer)

$$\mathcal{L}_C = \frac{1}{2} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

Center loss: direct clustering of features around centers

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_S + \lambda \mathcal{L}_C \\ &= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2\end{aligned}$$

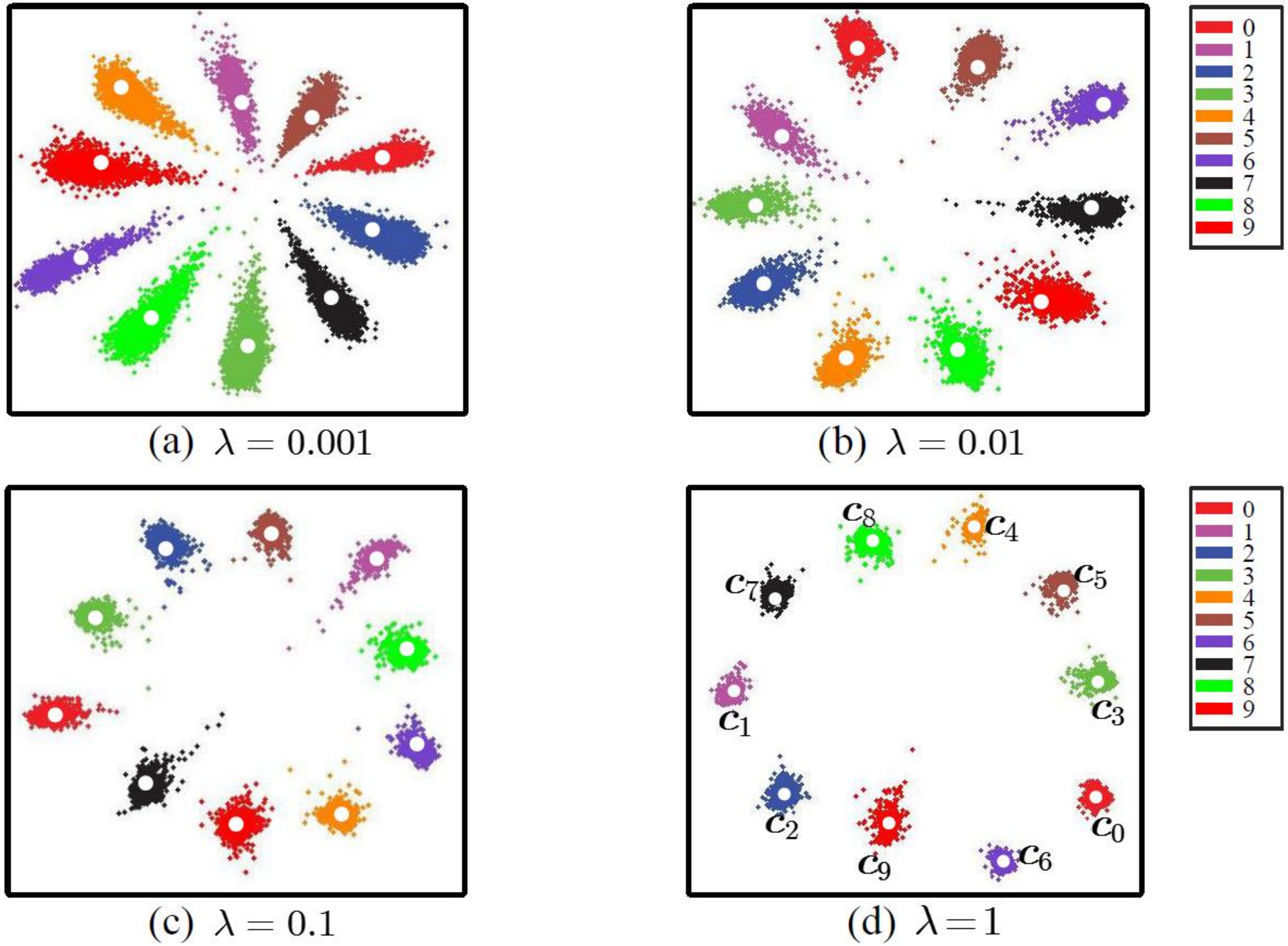
Total loss

Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image Similarity: Loss functions: **Center Loss**

MNIST handwritten digits 0-9

Feature clustering for softmax+center loss



Wen et al, "A Discriminative Feature Learning Approach for Deep Face Recognition", 2016

Image Similarity: Loss functions: **Contrastive loss**

$$L_i(X_1, X_2) = (1 - Y) D_{X_1, X_2}^2 + Y \max\left(0, m - D_{X_1, X_2}\right)^2$$

Hadsell et al, "Dimensionality Reduction by Learning an Invariant Mapping", 2006

Image Similarity: Loss functions: **Contrastive loss**

X_1 and X_2 images to compare

$$L_i\left(\boxed{X_1, X_2}\right) = (1 - Y)D^2_{X_1,X_2} + Y\max\left(0, m - \boxed{D_{X_1,X_2}}\right)^2$$

D – distance between image embeddings

Hadsell et al, "Dimensionality Reduction by Learning an Invariant Mapping", 2006

Image Similarity: Loss functions: **Contrastive loss**

X_1 and X_2 images to compare

margin

$$L_i\boxed{(X_1, X_2)} = (1 - \boxed{Y})D^2_{X_1,X_2} + Y\max(0, \boxed{m} - \boxed{D_{X_1,X_2}})^2$$

Y=0 if X_1 and X_2 are similar
Y=1 if X_1 and X_2 dissimilar

D – distance between image embeddings

Hadsell et al, "Dimensionality Reduction by Learning an Invariant Mapping", 2006

Image Similarity: Loss functions: **Contrastive loss**

X_1 and X_2 images to compare

margin

$$L_i\left(\boxed{X_1, X_2}\right) = (1 - \boxed{Y})D^2_{X_1,X_2} + Y\max\left(0, \boxed{m} - \boxed{D_{X_1,X_2}}\right)^2$$

D – distance between image embeddings

Y=0 if X_1 and X_2 are similar $\quad L_i(X_1, X_2) = D^2$

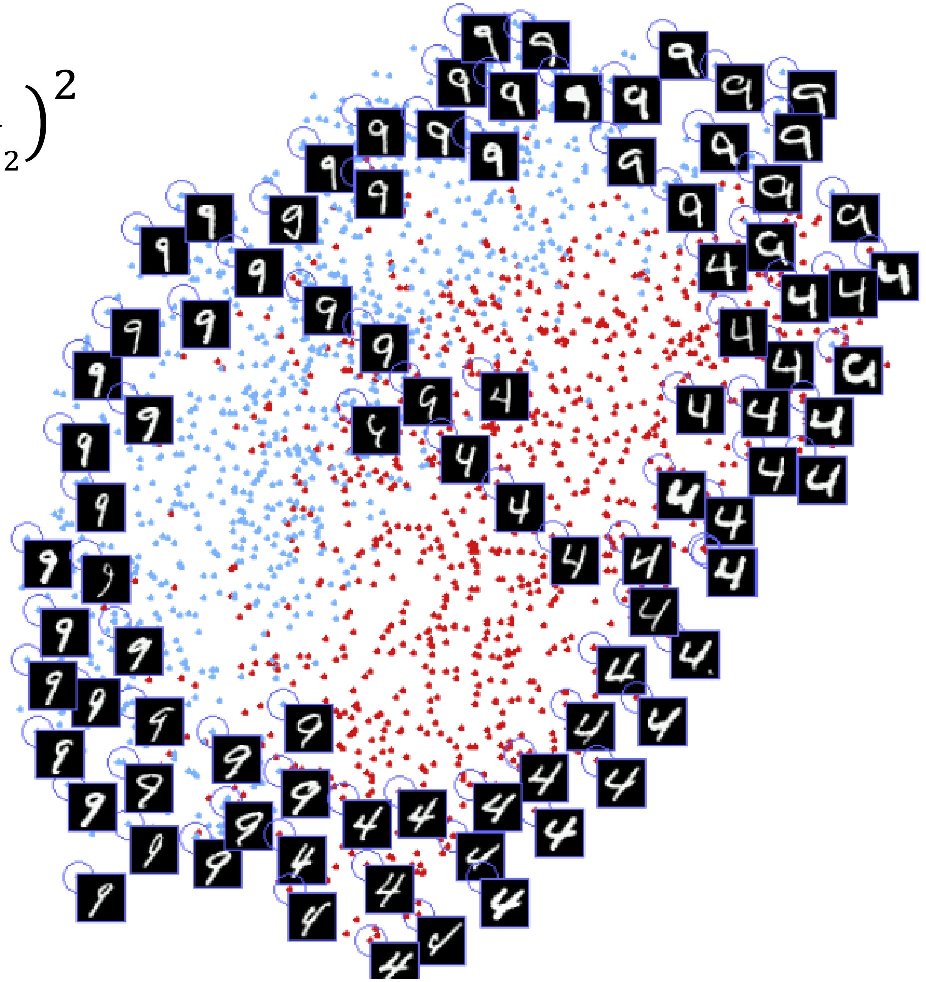
Y=1 if X_1 and X_2 dissimilar $\quad L_i(X_1, X_2) = \max(0, m - D)^2$

Hadsell et al, "Dimensionality Reduction by Learning an Invariant Mapping", 2006

Image Similarity: Loss functions: **Contrastive loss**

$$L_i(X_1, X_2) = (1 - Y)D^2_{X_1, X_2} + Y\max\left(0, m - D_{X_1, X_2}\right)^2$$

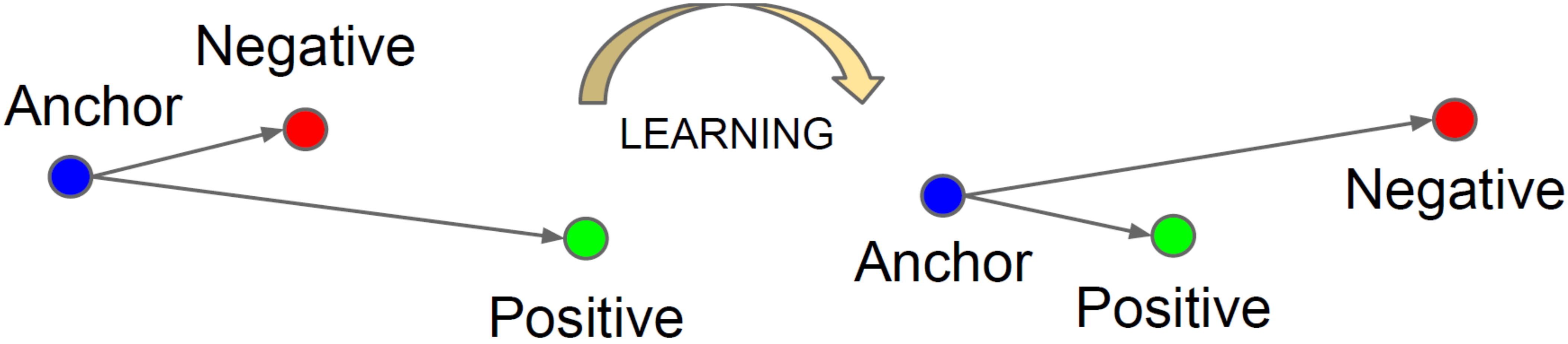
MNIST handwritten digits 0-9



test

Hadsell et al, "Dimensionality Reduction by Learning an Invariant Mapping", 2006

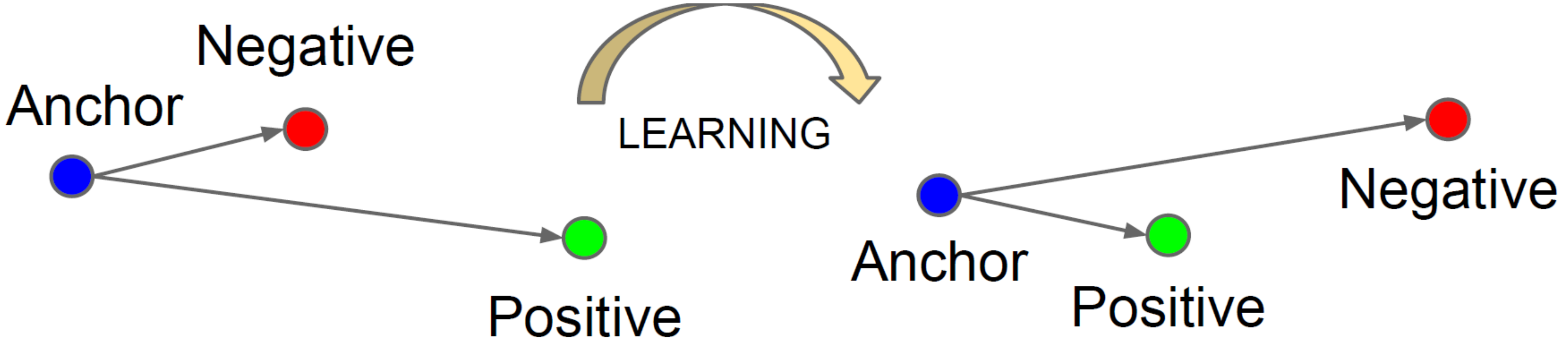
Image Similarity: Loss functions: **Triplet Loss**



$$L_i(a,p,n) = \max(0, D_{a,p}^2 - D_{a,n}^2 + m)$$

Schroff et al, "FaceNet: A Unified Embedding for Face Recognition and Clustering", 2015

Image Similarity: Loss functions: **Triplet Loss**



margin

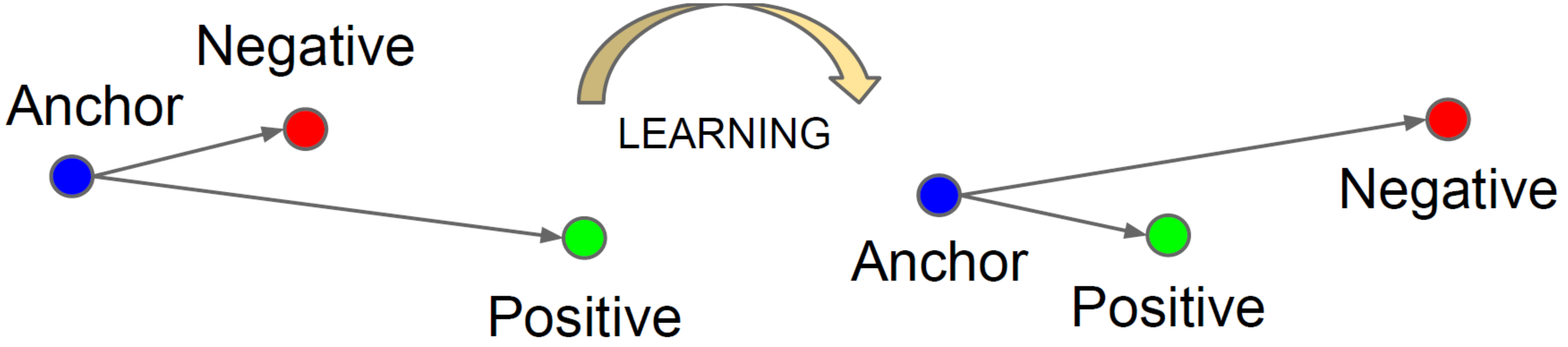
$$L_i(\boxed{a, p, n}) = \max(0, \boxed{D_{a,p}^2} - \boxed{D_{a,n}^2} + \boxed{m})$$

a – anchor
p – positive
n – negative

distances between image embeddings

Schroff et al, "FaceNet: A Unified Embedding for Face Recognition and Clustering", 2015

Image Similarity: Loss functions: **Triplet Loss**



$$L_i(a,p,n) = \max(0, D_{a,p}^2 - D_{a,n}^2 + m)$$

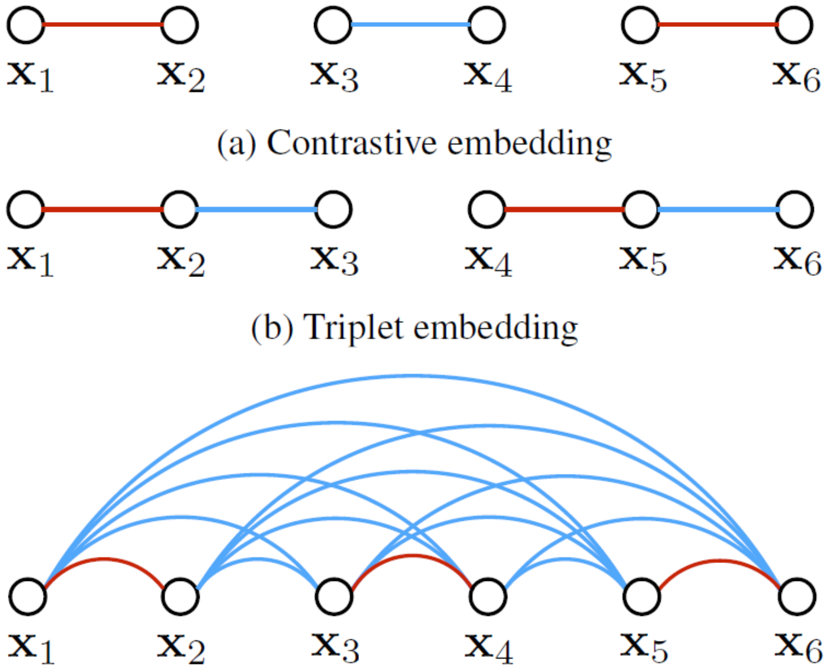
Hard negative mining:
for anchor-positive pair
search for difficult negative:

$$\min_n \left(D_{a,n}^2\right)$$

Schroff et al, "FaceNet: A Unified Embedding for Face Recognition and Clustering", 2015

Image Similarity: Loss functions: **Lifted Structured Loss**

Automatic hard negative mining within batch



(a) Contrastive embedding

(b) Triplet embedding

(c) Lifted structured embedding

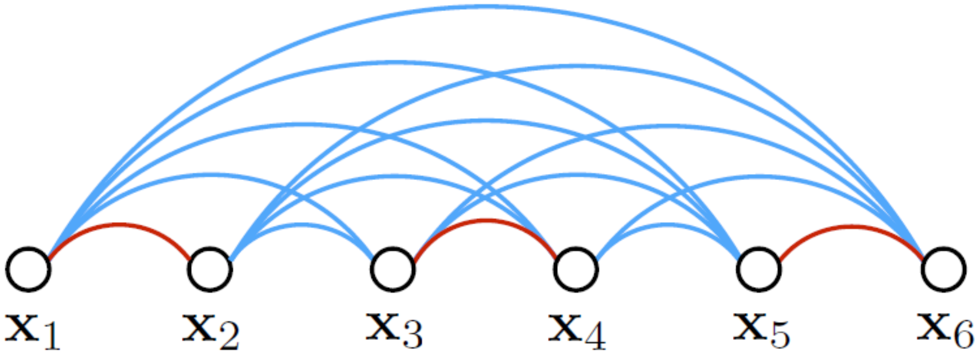
Training batch with six examples. Red edges and blue edges represent similar and dissimilar examples respectively.

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

Image Similarity: Loss functions: **Lifted Structured Loss**

$$\tilde{J}_{i,j} = \log\left(\sum_{(i,k)\in N} \exp(m - D_{i,k}) + \sum_{(j,l)\in N} \exp(m - D_{j,l})\right) + D_{i,j}$$

$$L_{batch} = \frac{1}{2|P|} \sum_{(i,j)\in P} max(0,\ \tilde{J}_{i,j})^2$$



(c) Lifted structured embedding

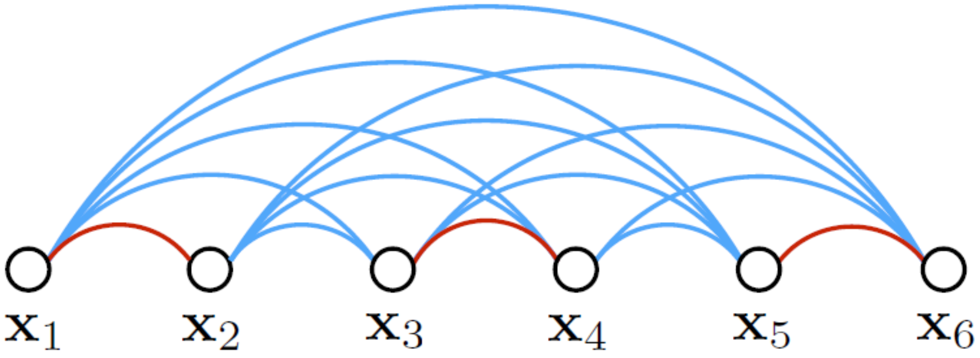
Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

Image Similarity: Loss functions: **Lifted Structured Loss**

$$\tilde{J}_{i,j} = \log\left(\sum_{(i,k)\in N} \exp(m - D_{i,k}) + \sum_{(j,l)\in N} \exp(m - D_{j,l})\right) + D_{i,j}$$

$$L_{batch} = \frac{1}{2|P|} \sum_{\boxed{(i,j)\in P}} max(0,\ \tilde{J}_{i,j})^2$$

all positive pairs
within batch



(c) Lifted structured embedding

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

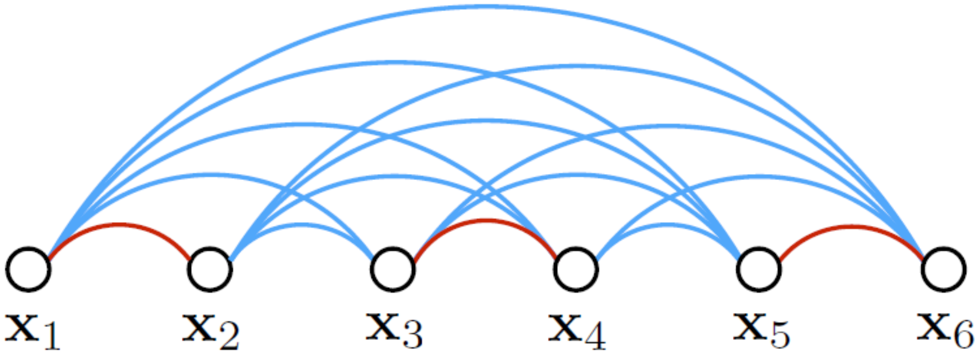
Image Similarity: Loss functions: **Lifted Structured Loss**

margin

$$\tilde{J}_{i,j} = \log\left(\sum_{(i,k)\in N} \exp(\boxed{m} - D_{i,k}) + \sum_{(j,l)\in N} \exp(m - D_{j,l})\right) + D_{i,j}$$

$$L_{batch} = \frac{1}{2|P|} \sum_{\boxed{(i,j)\in P}} max(0,\ \tilde{J}_{i,j})^2$$

all positive pairs
within batch



(c) Lifted structured embedding

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

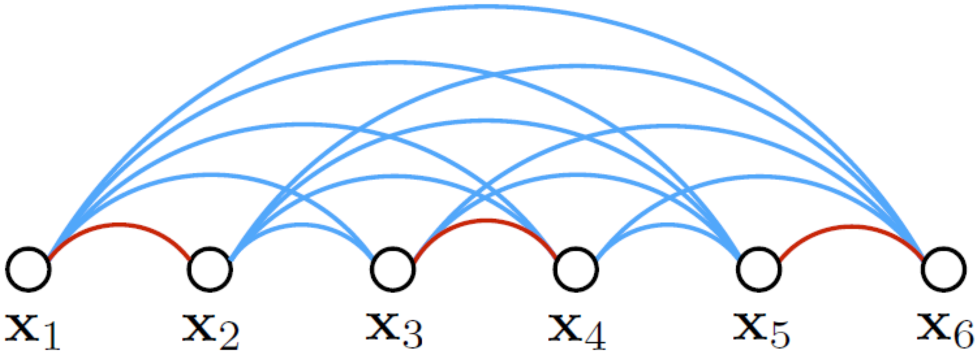
Image Similarity: Loss functions: **Lifted Structured Loss**

$$\tilde{J}_{i,j} = \log\left(\sum_{\boxed{(i,k)\in N}} \exp(m - D_{i,k}) + \sum_{\boxed{(j,l)\in N}} \exp(m - D_{j,l})\right) + D_{i,j}$$

all negative pairs for *i* all negative pairs for *j*

$$L_{batch} = \frac{1}{2|P|} \sum_{\boxed{(i,j)\in P}} max(0,\ \tilde{J}_{i,j})^2$$

all positive pairs within batch



(c) Lifted structured embedding

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

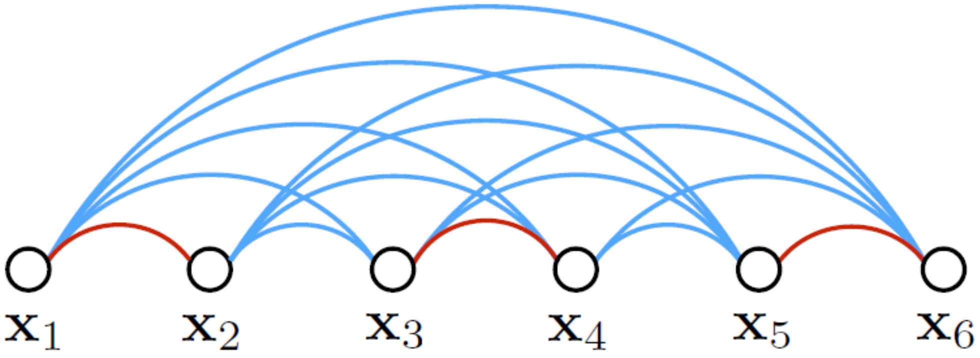
Image Similarity: Loss functions: **Lifted Structured Loss**

softmax ☺ gives hardest negative for *i* and *j*

$$\tilde{J}_{i,j} = \boxed{\log\left(\sum_{(i,k)\in N} \exp(m - D_{i,k}) + \sum_{(j,l)\in N} \exp(m - D_{j,l})\right)} + D_{i,j}$$

$$L_{batch} = \frac{1}{2|P|} \sum_{\boxed{(i,j)\in P}} max(0,\ \tilde{J}_{i,j})^2$$

all positive pairs within batch



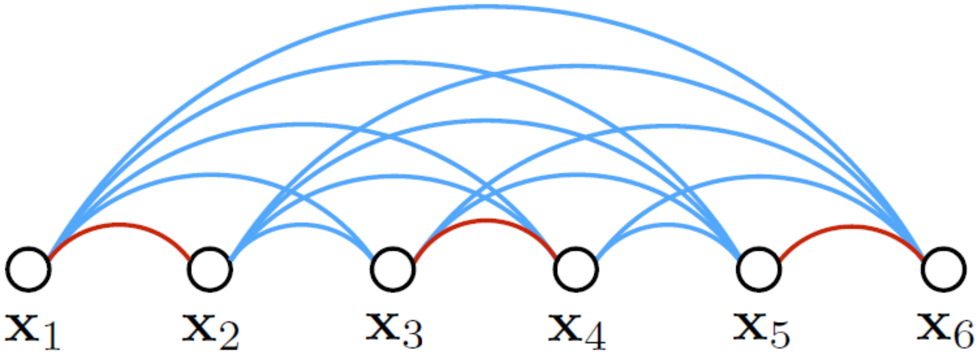
(c) Lifted structured embedding

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

Image Similarity: Loss functions: **Lifted Structured Loss**

$$L_{batch} \sim \frac{1}{2|P|} \sum_{(i,j)\in P} max(0,\ D_{i,j} - D_{i|j,hardest\ negative} + \mathrm{m})^2$$

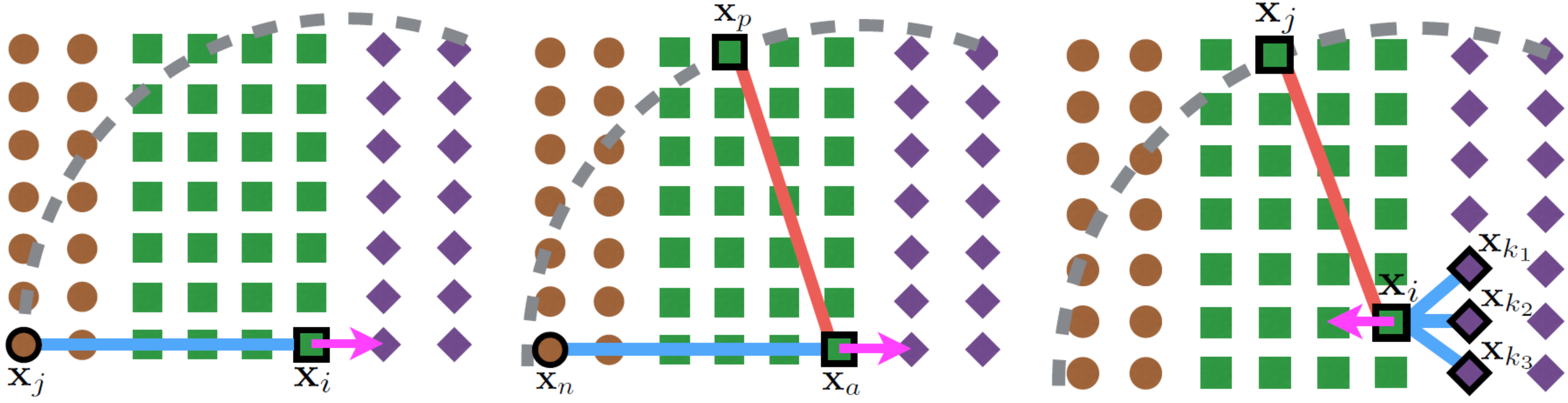
Similar to triplet loss but with automatic hard negative mining within batch!



(c) Lifted structured embedding

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

Image Similarity: Loss functions: **Lifted Structured Loss**



(a) Contrastive embedding (b) Triplet embedding (c) Lifted structured similarity

Illustration of failure modes of contrastive and triplet loss with randomly sampled training batch.

Song et al, "Deep Metric Learning via Lifted Structured Feature Embedding", 2015

ICCV 2017: Venice, Italy, Oct 22-29

Growth of paper submissions
~30%

Growth of participants
~200%

ICCV 2017: We are hiring

We are hiring ☺