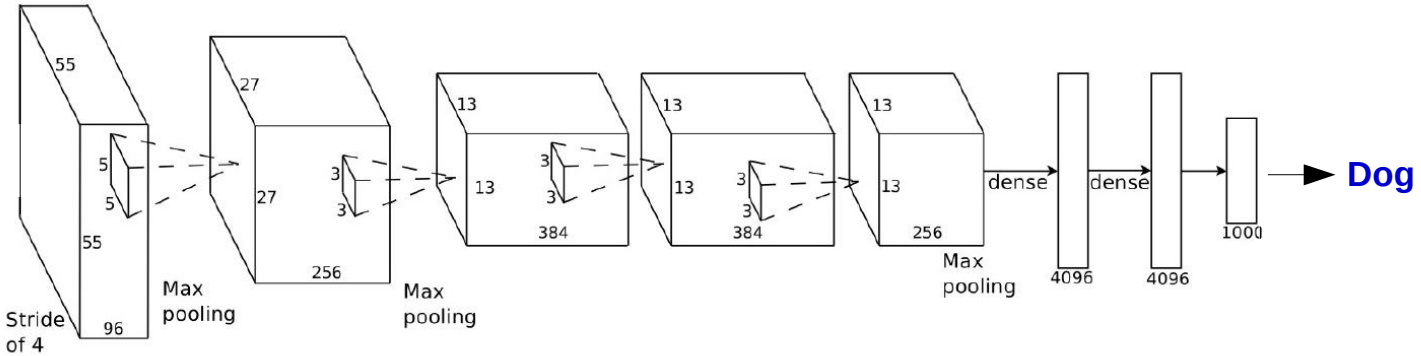
Contrastive Loss

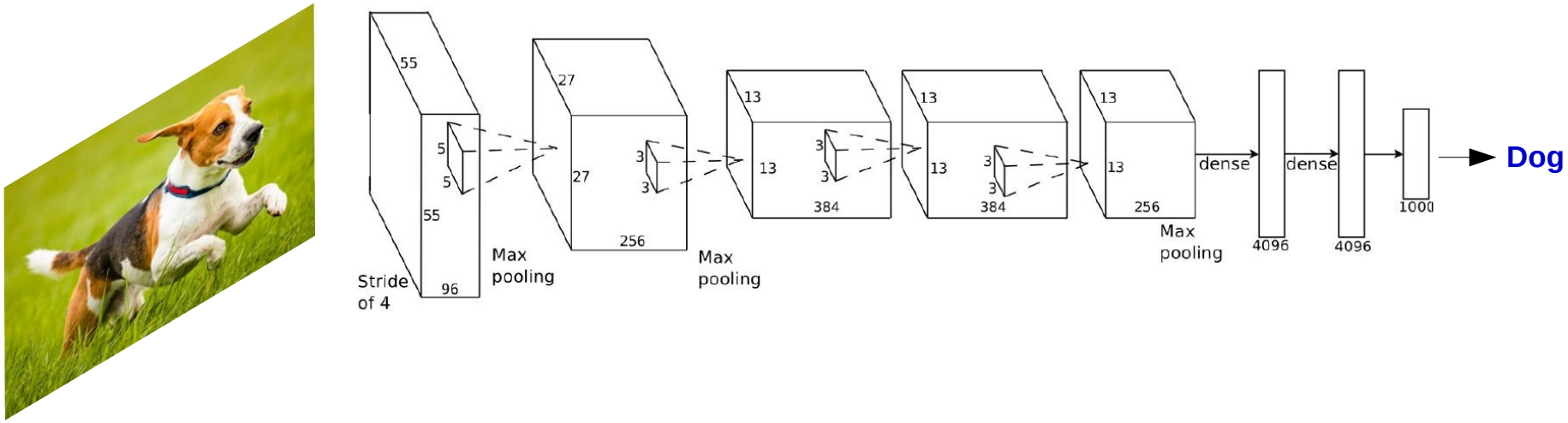
Alvaro Soto

Computer Science Department (DCC), PUC

Visual Multi-Way Classification



Visual Multi-Way Classification

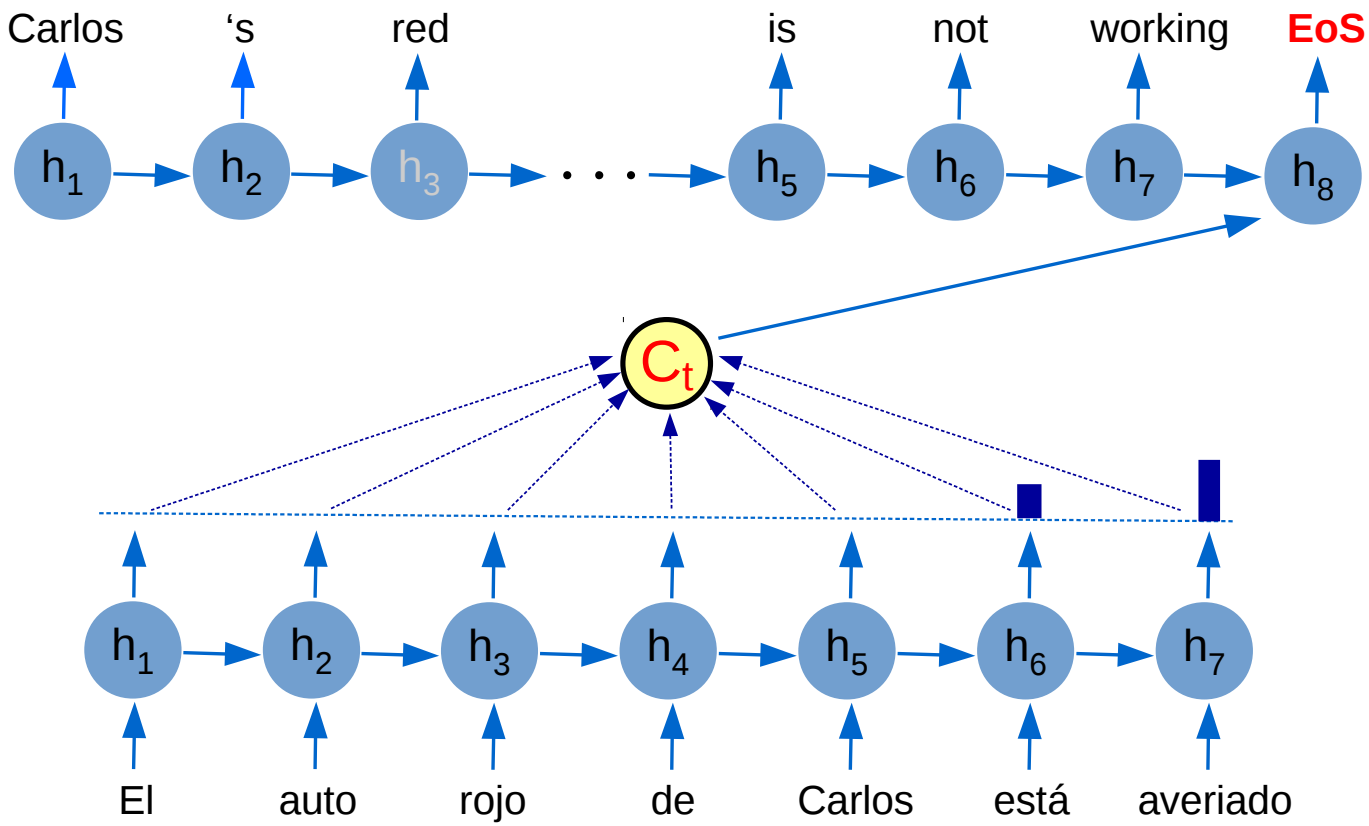


We train this network using cross-entropy, i.e., we fit weights so that the network aims to reproduce the distribution of labels in the training set.

$$H(p(x), q(x)) = -E_{x \sim p} \log q(x)$$

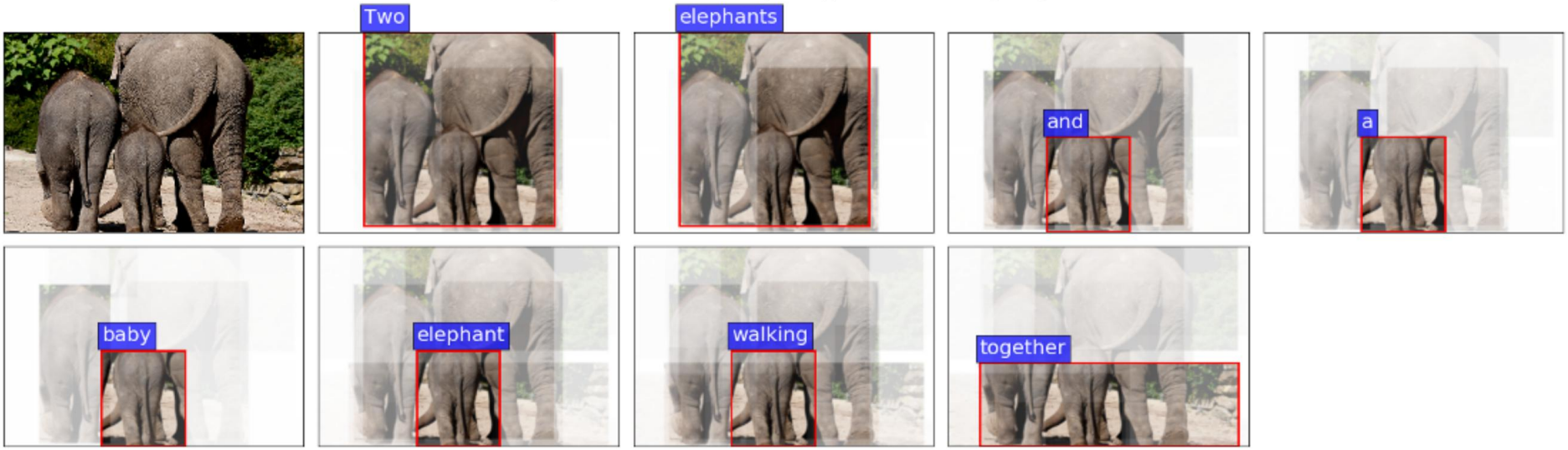
Cross-entropy: overhead of transmiting a signal defined by a pdf $p(x)$ using an encoding that considers a pdf $q(x)$.

Seq2Seq Translation



Multi-Modal Applications

Two elephants and a baby elephant walking together.



For some relevant applications, multi-way classification or Seq2Seq transformation are not the best options.

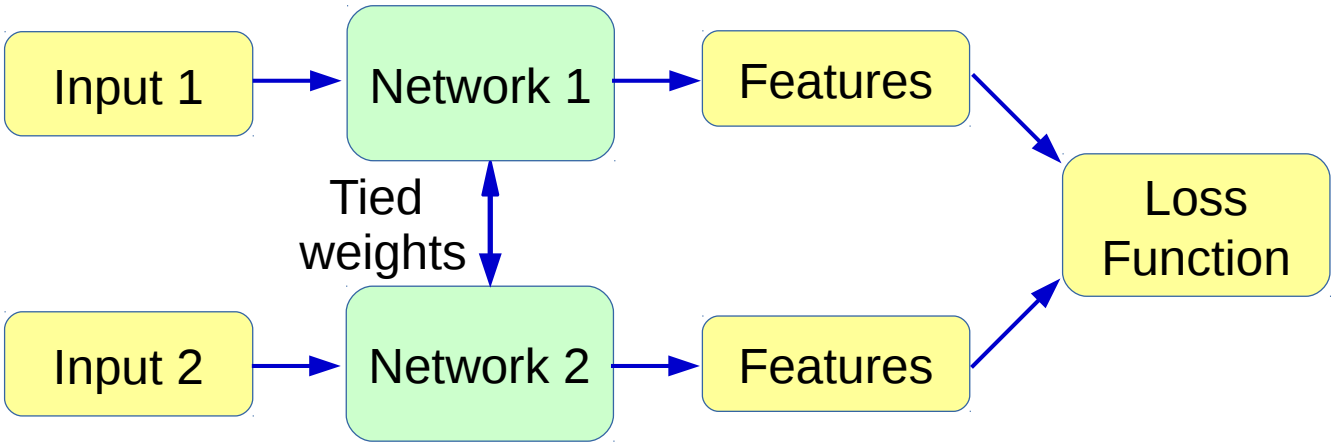
In this class, we will discuss alternative training mechanisms and loss functions that can lead to more robust models.

Example: Image comparison

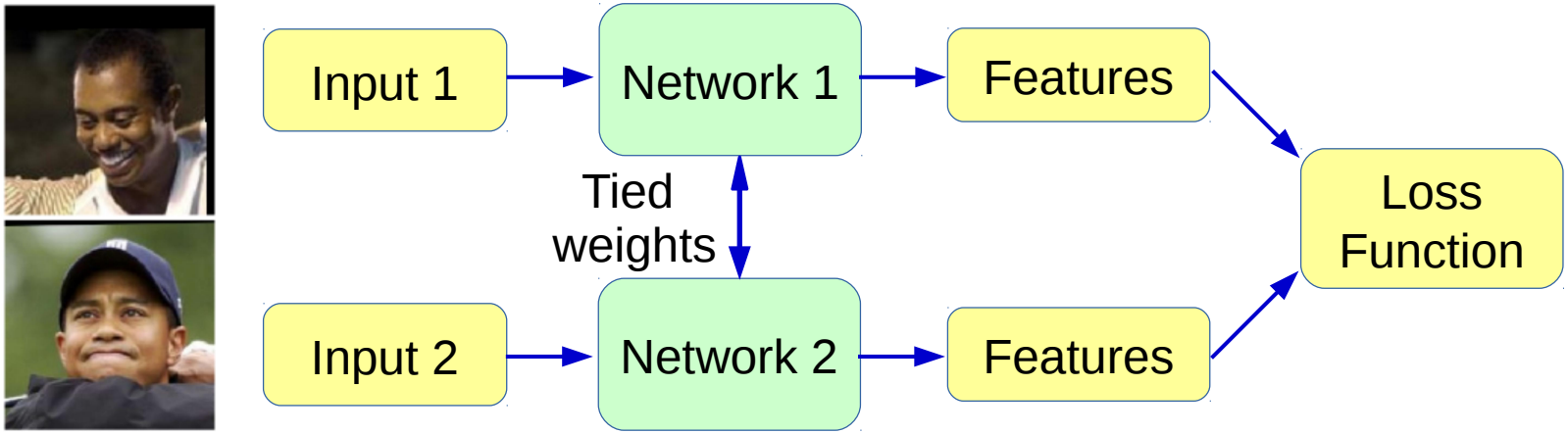
Are these pairs of images from the same person?




Siamese Network



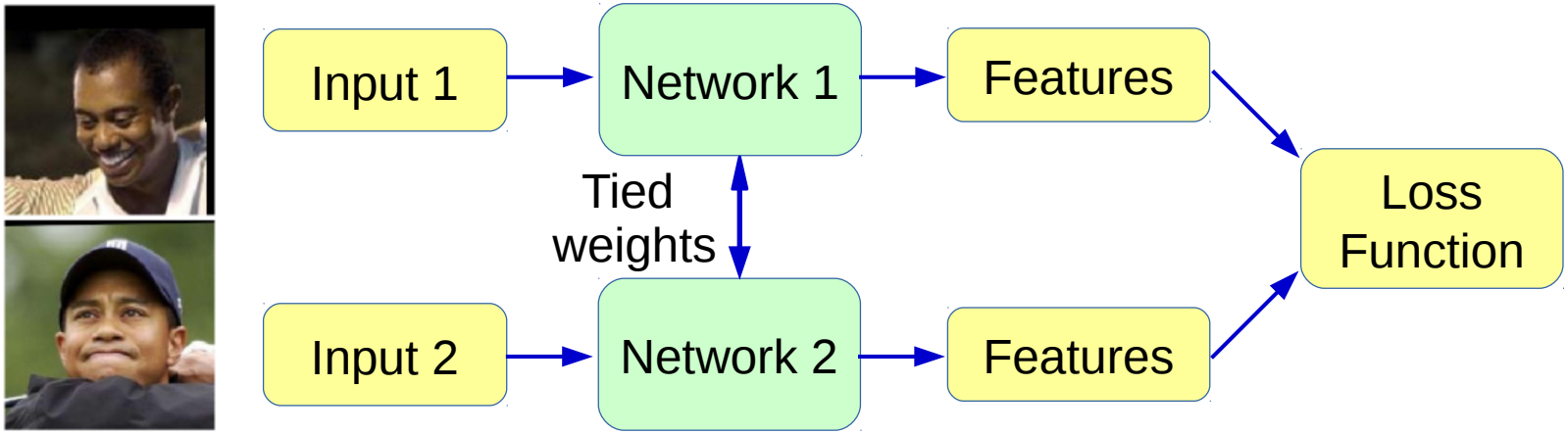
Siamese Network



Idea:

If input images are from same class decrease distance between them; otherwise increase distance.

Siamese Network



Idea:

If input images are from same class decrease distance between them; otherwise increase distance.

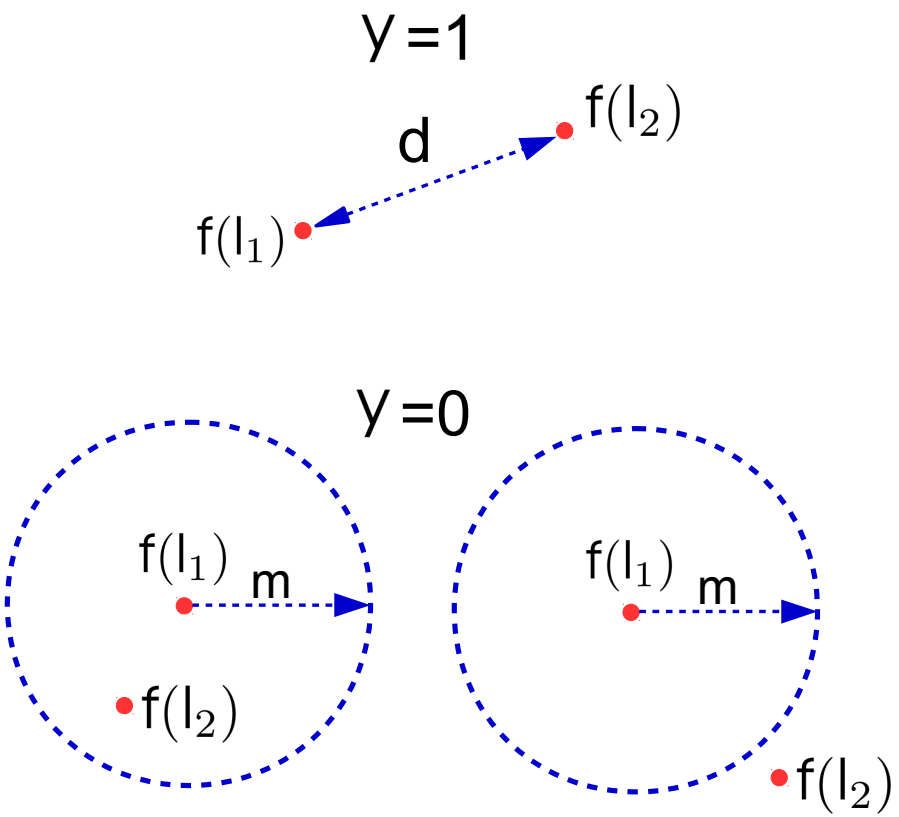
Pairwise Ranking Loss:

$$L(f(I_1), f(I_2), y) = y \, \|f(I_1) - f(I_2)\| + (1 - y)\max\{0, m - \|f(I_1) - f(I_2)\|\}$$

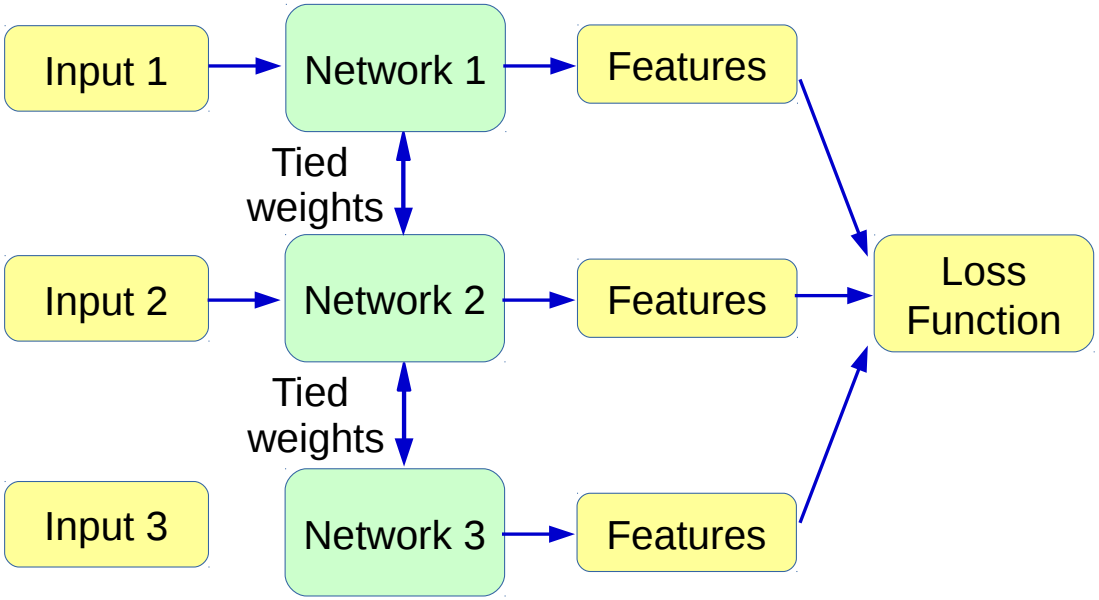
y: binary class label. $y = 1$ if images are from the same class.

Pairwise Ranking Loss:

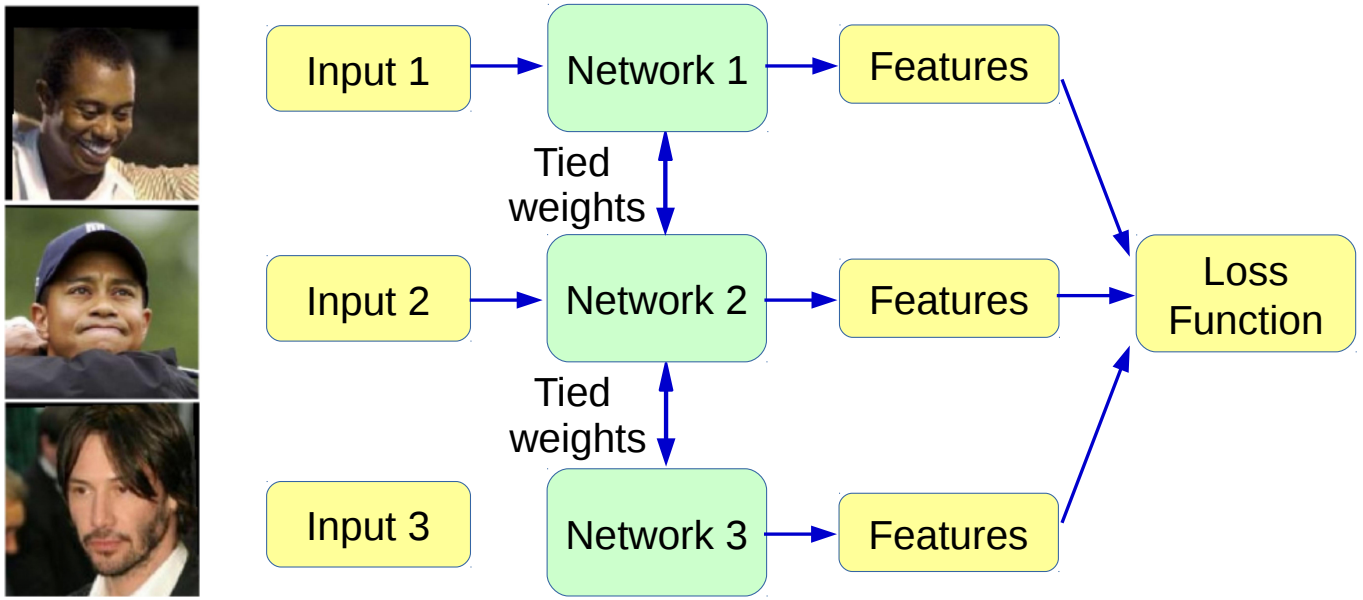
$$L(f(I_1), f(I_2), y) = y\,\|f(I_1) - f(I_2)\| + (1-y)\max\{0, m - \|f(I_1) - f(I_2)\|\}$$



Triplet Network



Triplet Network

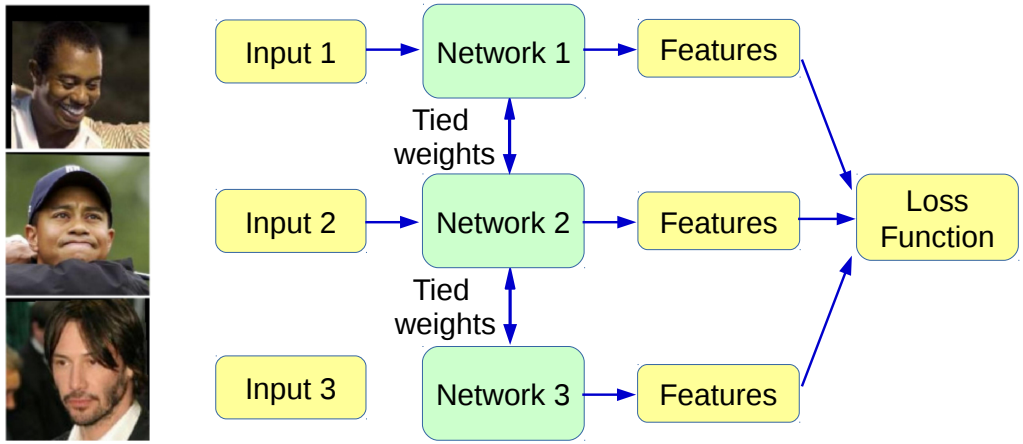


Triplet Ranking Loss:

$$L(f(I_1), f(I_2), f(I_3), y) = \max\left\{0, m - \left\{\left\|f(I_1) - f(I_3)^-\right\| - \left\|f(I_1) - f(I_2)^+\right\|\right\}\right\}$$

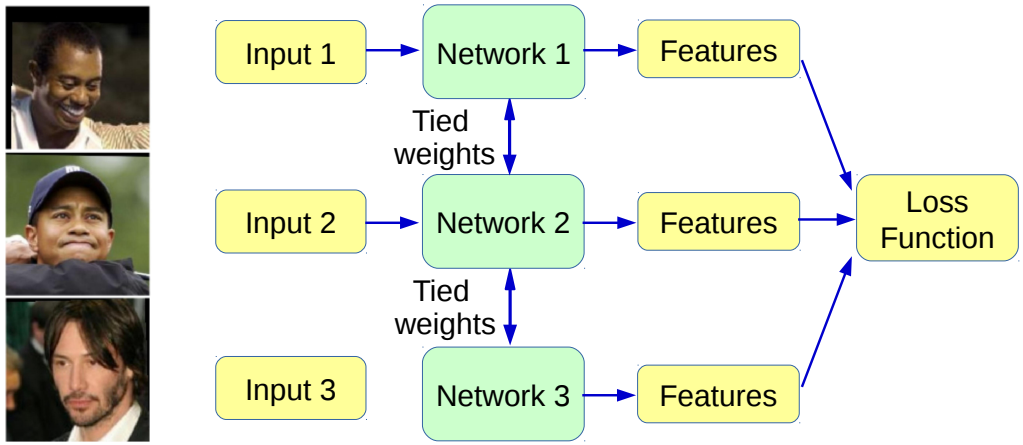
Triplet Ranking Loss:

$$L(f(I_1), f(I_2), f(I_3), y) = \max\left\{0, m - \left\{\left\|f(I_1) - f(I_3)^-\right\| - \left\|f(I_1) - f(I_2)^+\right\|\right\}\right\}$$



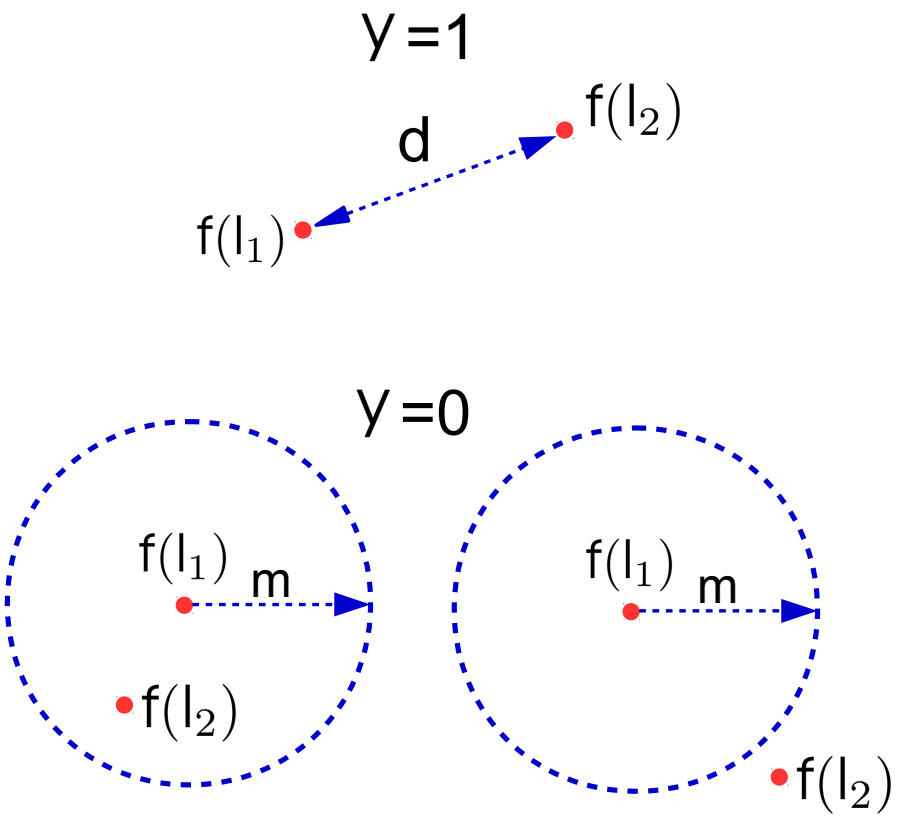
Triplet ranking loss is a contrastive loss

$$L(f(I_1), f(I_2), f(I_3), y) = \max\left\{0, m - \left\{\left\|f(I_1) - f(I_3)^-\right\| - \left\|f(I_1) - f(I_2)^+\right\|\right\}\right\}$$



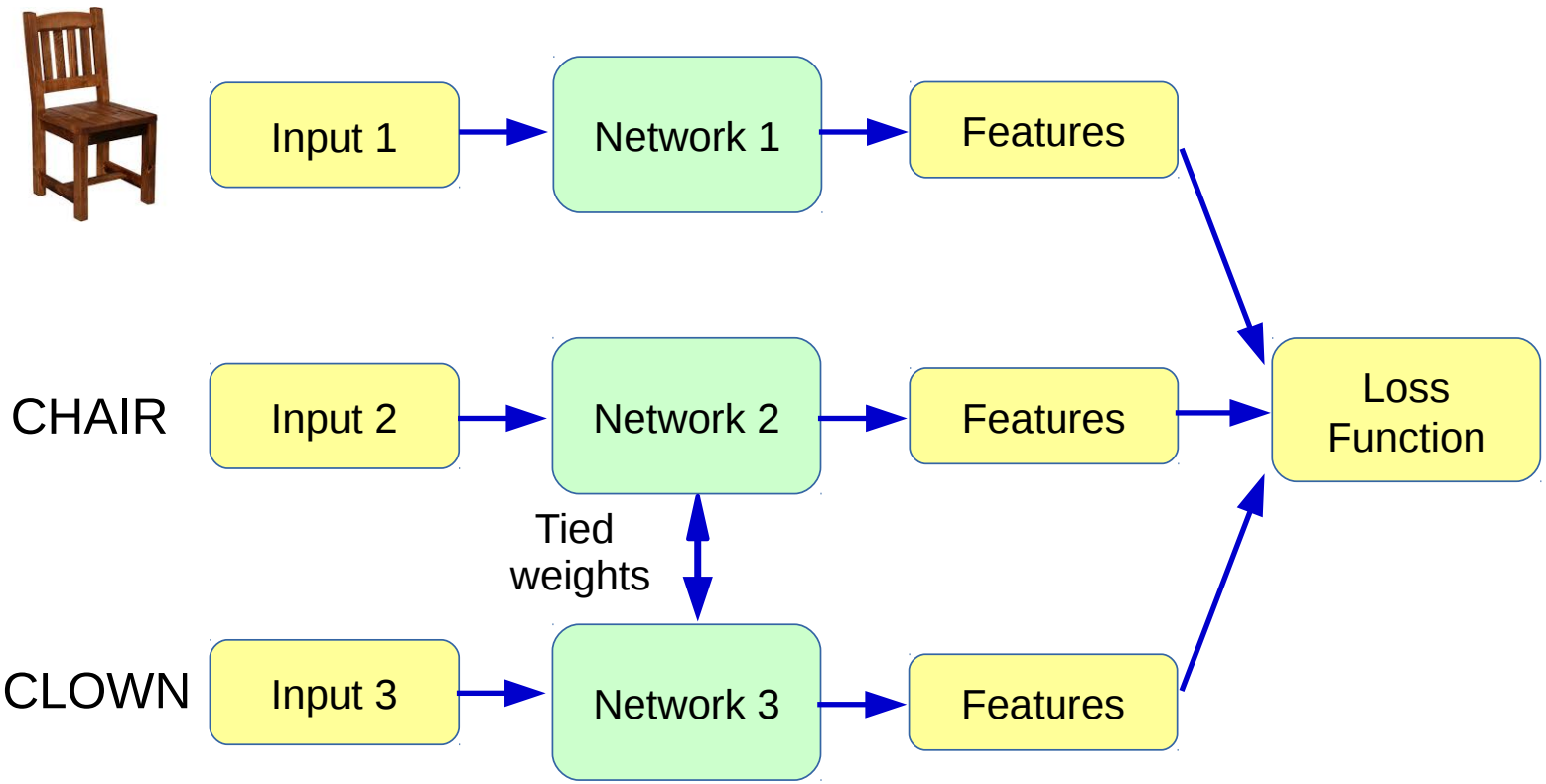
- Contrastive loss: contrast distance between a pair of examples from same class with respect to an example from a different class
- GANs are based on a constrastive loss, but they integrate a hard mining mechanism
- Hard Mining?.

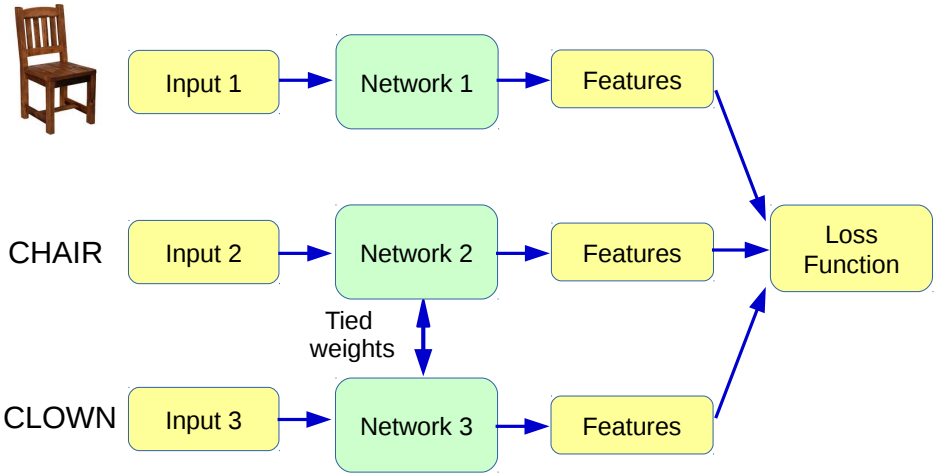
Hard Mining

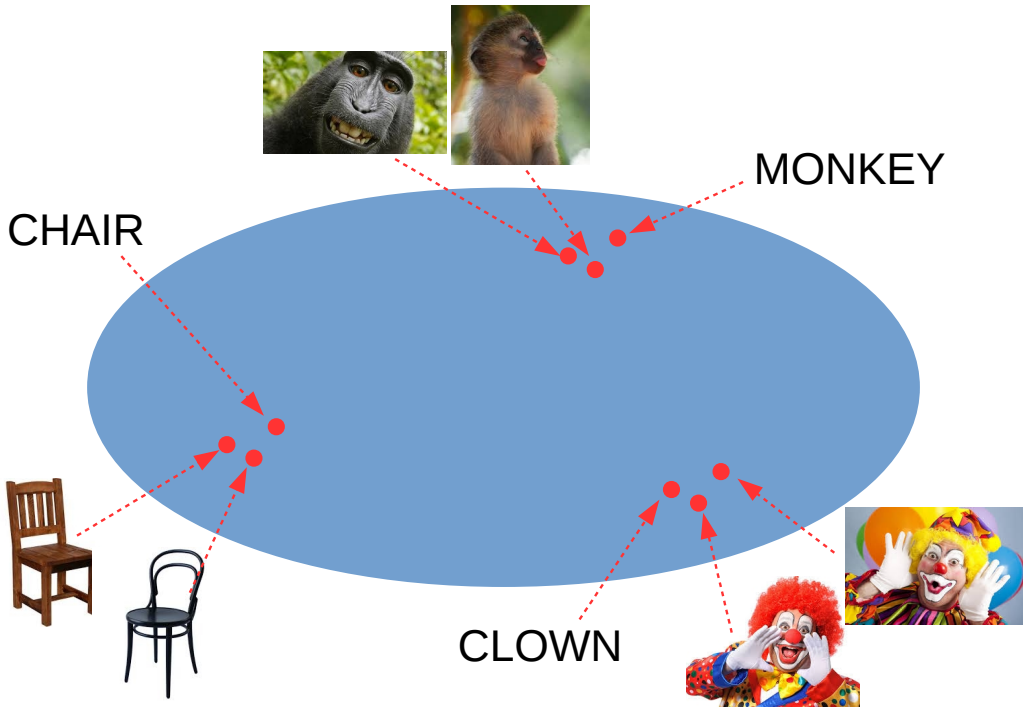


Hard Mining

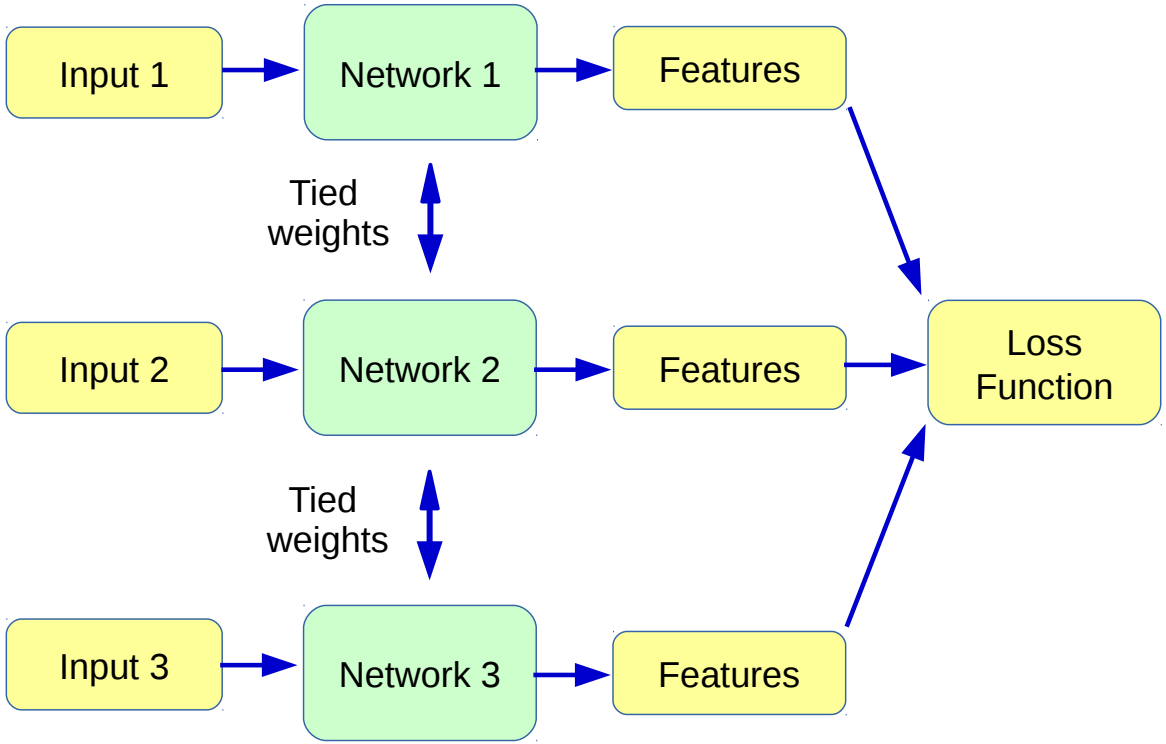
Application Example: Multimodal Embeedings




Input 1
Network 1
Features
CHAIR
Input 2
Network 2
Features
Loss Function
Tied weights
CLOWN
Input 3
Network 3
Features


MONKEY
CHAIR
CLOWN

Application Example: Robot Localization

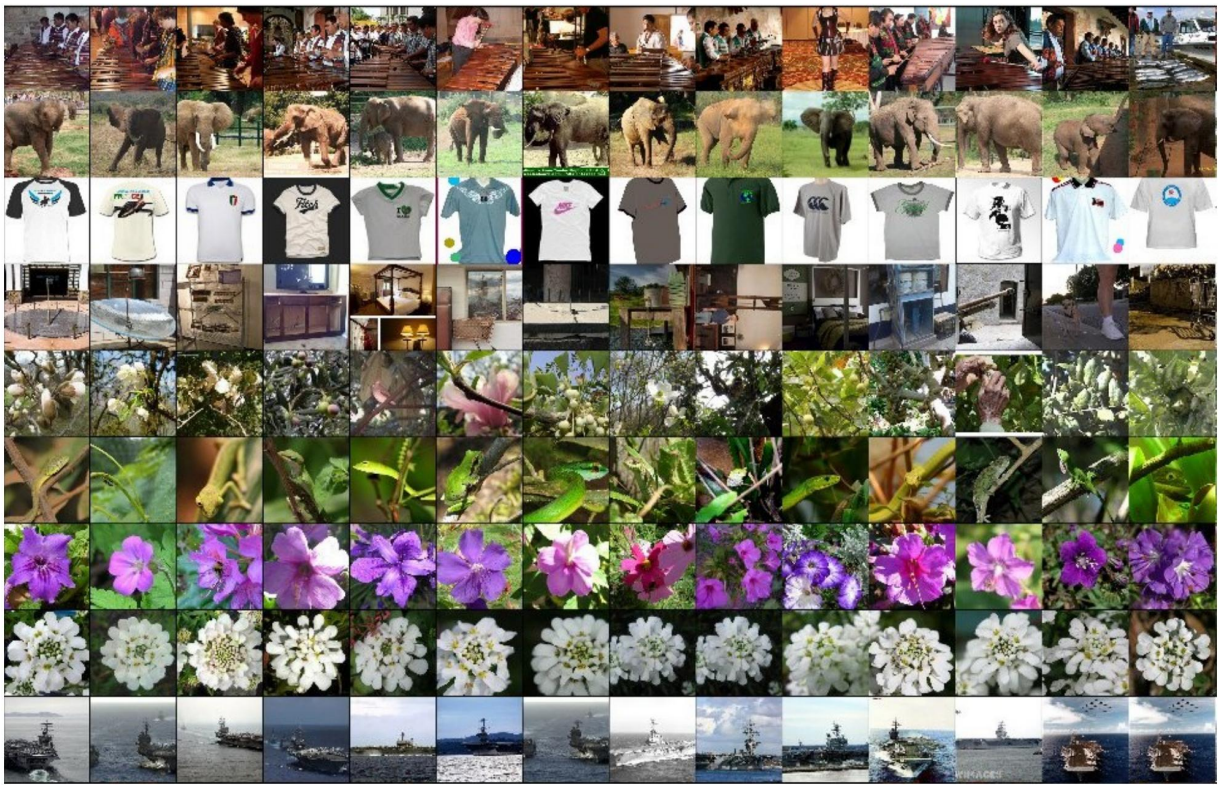


Distributed Representations

Triplet Ranking Loss:

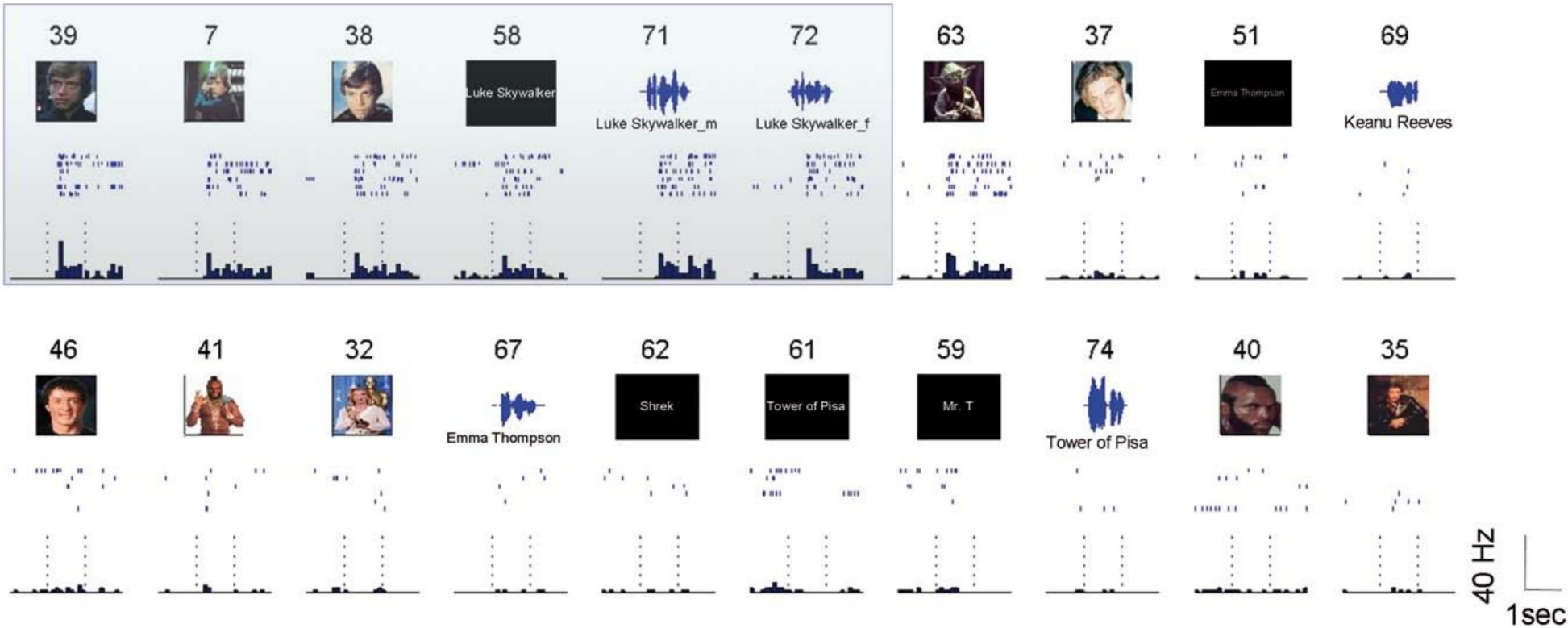
$$L(f(I_1), f(I_2), f(I_3), y) = \max\left\{0, m - \left\{\left\|f(I_1) - f(I_3)^-\right\| - \left\|f(I_1) - f(I_2)^+\right\|\right\}\right\}$$

AlexNet Embeddings



Learning a semantic distance metric

Distributed Representations in Human Brain?



Generative Adversarial Networks (GANs)

Generative Models

Learn to generate samples from same distribution that input data

- Goal: learn a model such that $P_{model}(x) \simeq P_{data}(x)$
- In other words, the goal is density estimation
- In contrast to traditional density estimation techniques, we will focus on high dimensional spaces. Ex. images.

Radford et al, ICLR 2016

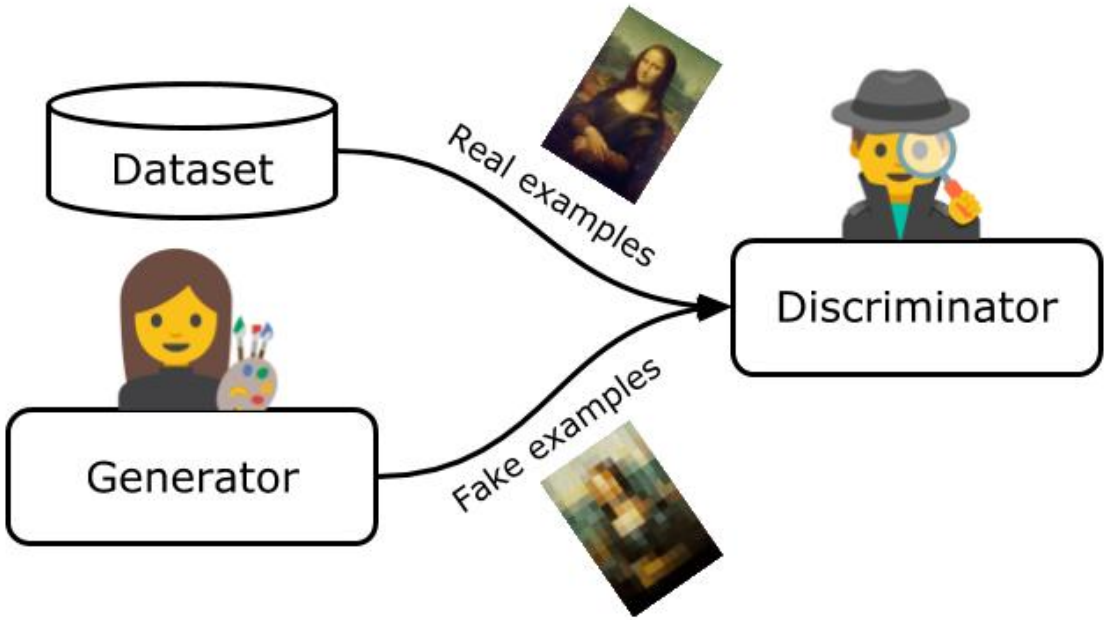
Generative Models

Learn to generate samples from same distribution that input data

- Two main approaches:
  - Explicit density estimation: explicitly find a model to define p(x)
  - Implicit density estimation: learn a model that can sample from p(x) w/o explicitly defining p(x).
- Several possible techniques:
  - PixelRNNs/CNNs
  - Autoencoders
  - Gans
  - Others ...

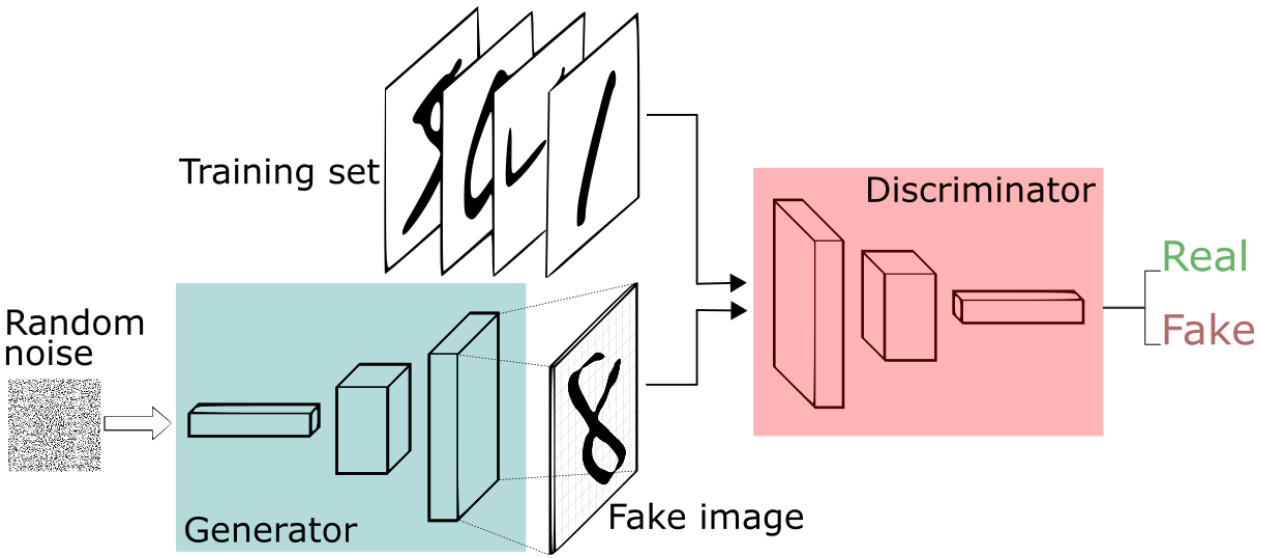
GANs

- GANs: don't work with any explicit density function
- It uses a game-theoretic approach to transform an unsupervised task (density estimation) into a supervised learning task
- Specifically, they use a 2-player zero-sum game: generator and discriminator.



GANs

- GANs train two different networks:
  - Generator network tries to produce realistic-looking samples
  - Discriminator network tries to figure out whether an image came from the training set or the generator network.



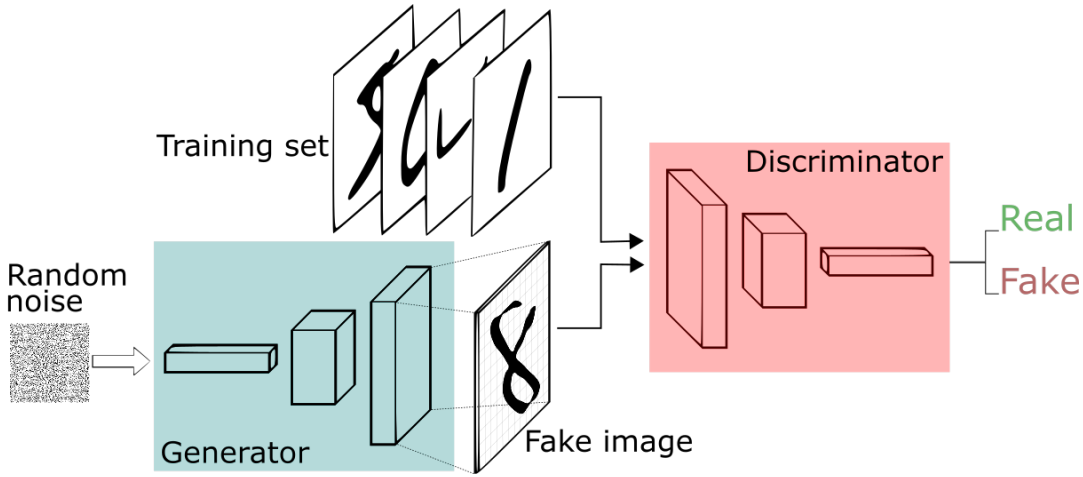
- Generator tries to fool the discriminator
- Discriminator tries to discover fakes from the generator.

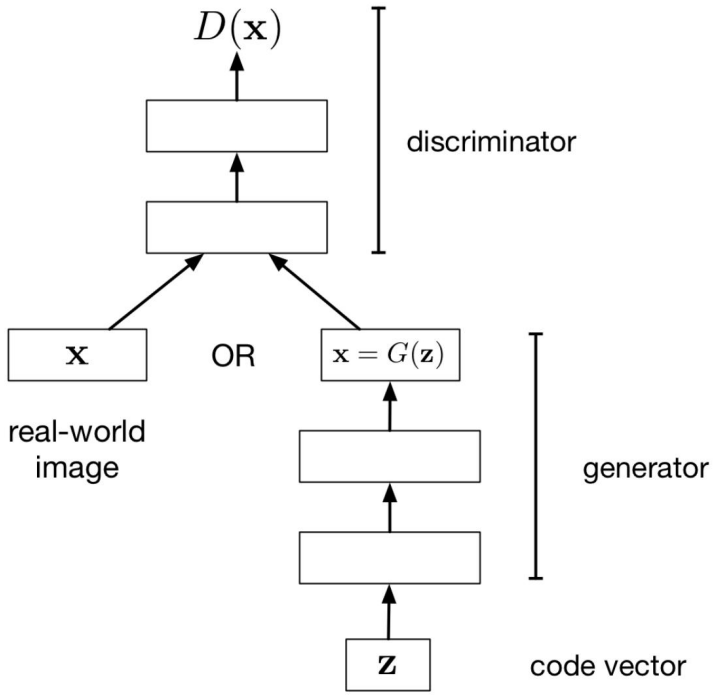
Loss Function

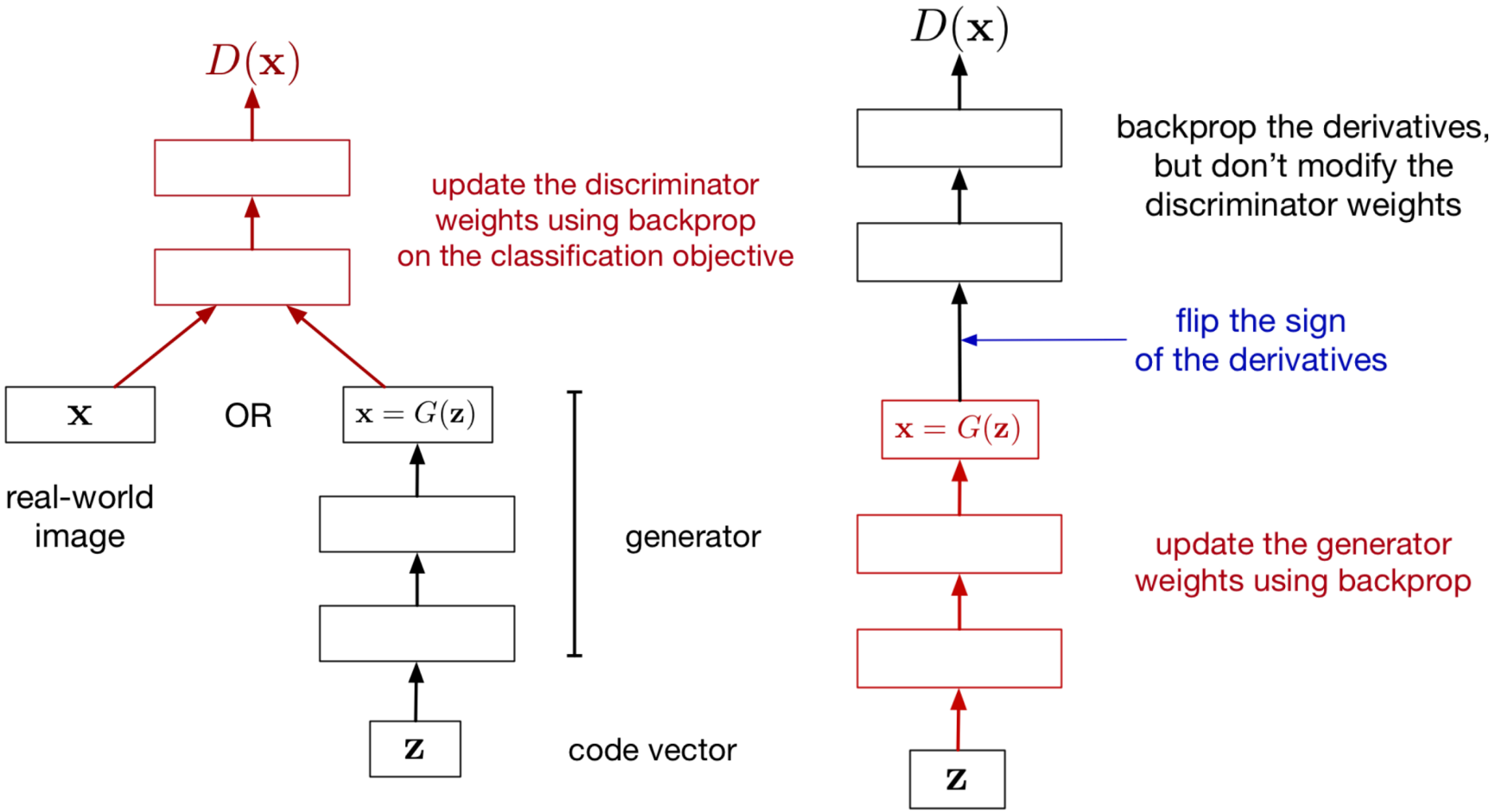
$$\min_{G}\max_{D} V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

Relevant Term: $D(G(z))$

- Discriminator: wants to minimize $D(G(z))$, i.e., classify with high confidence that the sample is coming from the fake distribution
- Generator: wants to maximize $D(G(z))$, i.e., fool the discriminator by making it to believe that sample z is from real distribution.

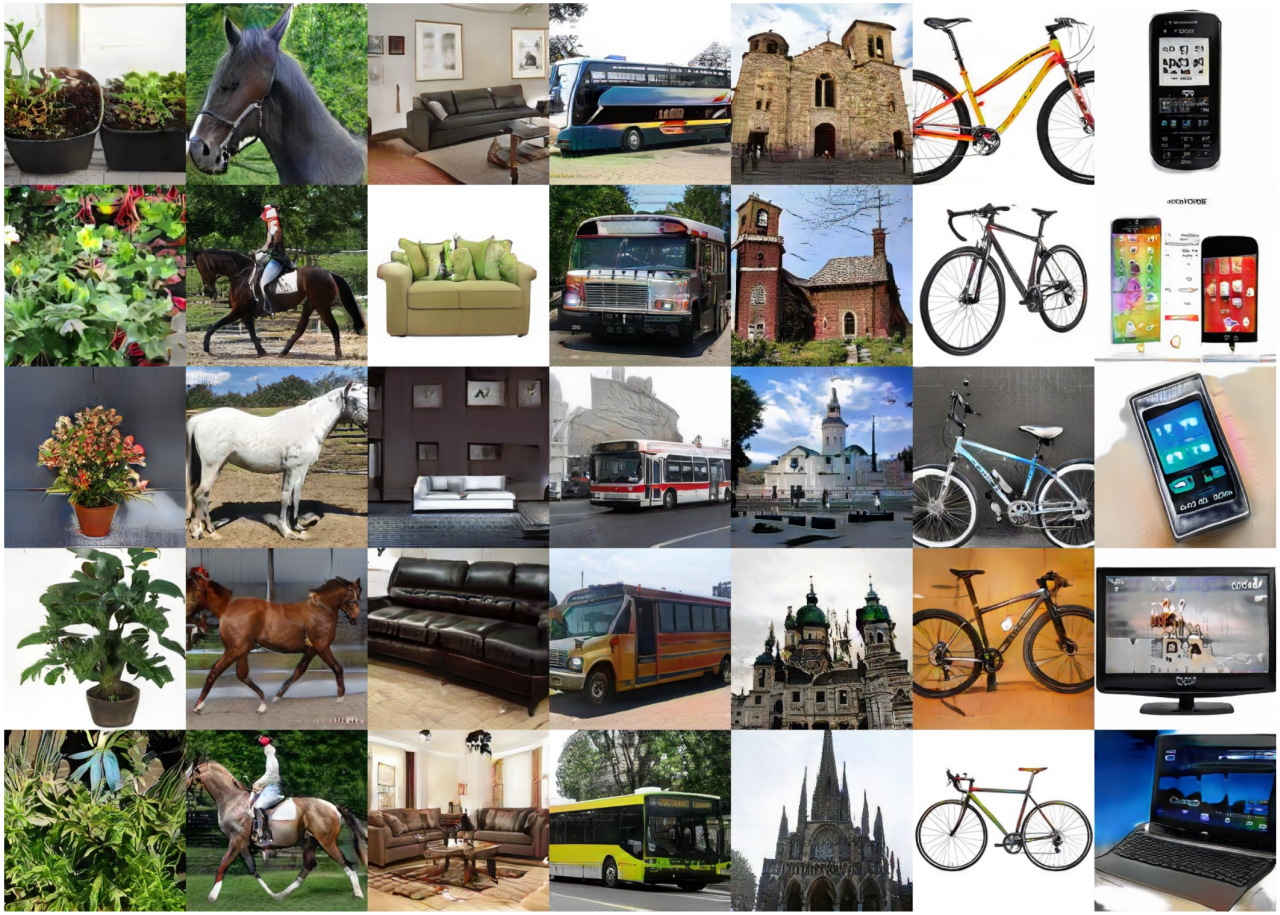

Training set
Discriminator
Real
Fake
Random
noise
Generator
Fake image


$D(\mathbf{x})$
discriminator
$\mathbf{x}$
OR
$\mathbf{x} = G(\mathbf{z})$
real-world
image
generator
$\mathbf{z}$
code vector


$D(\mathbf{x})$
update the discriminator
weights using backprop
on the classification objective
$\mathbf{x}$
OR
$\mathbf{x} = G(\mathbf{z})$
real-world
image
generator
$\mathbf{z}$
code vector
$D(\mathbf{x})$
backprop the derivatives,
but don't modify the
discriminator weights
flip the sign
of the derivatives
$\mathbf{x} = G(\mathbf{z})$
update the generator
weights using backprop
$\mathbf{z}$

1024x1024 images generated using the CELEBA-HQ dataset Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation.

256x256 images generated from LSUN bedroom category
Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation.

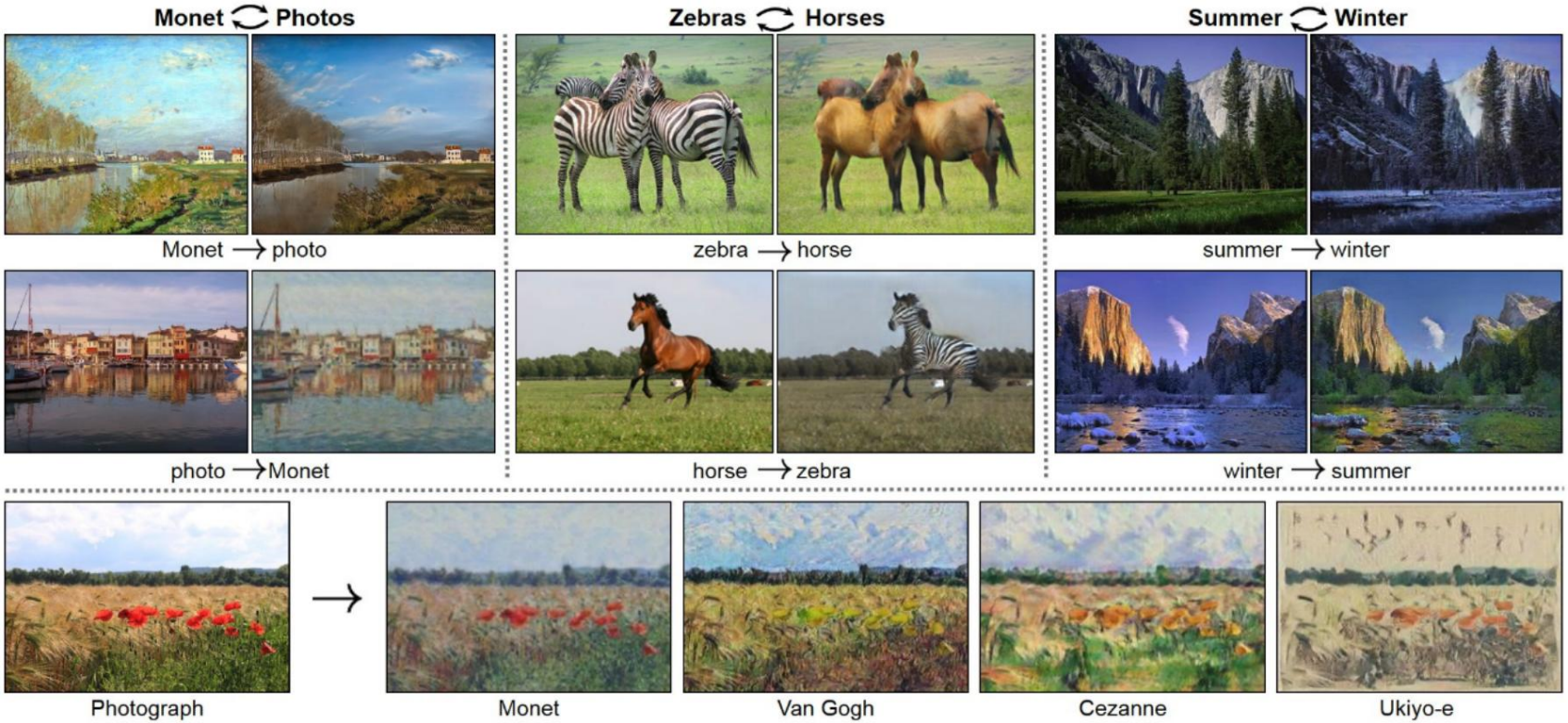


256x256 images generated from different LSUN categories
Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation.

Cycle GAN

Style Transfer

GAN transfers an image from one domain to the style of a second domain



Cycle GAN

Real zebra image

The discriminator tries to distinguish generated zebra images from real ones

D

Discriminator loss: GAN generator objective, i.e. negative log probability D assigns to the sample being real

Reconstruction loss: squared error between the original image and the reconstruction

Input image (real horse image)

G_1

Generator 1 learns to map from horse images to zebra images while preserving the structure

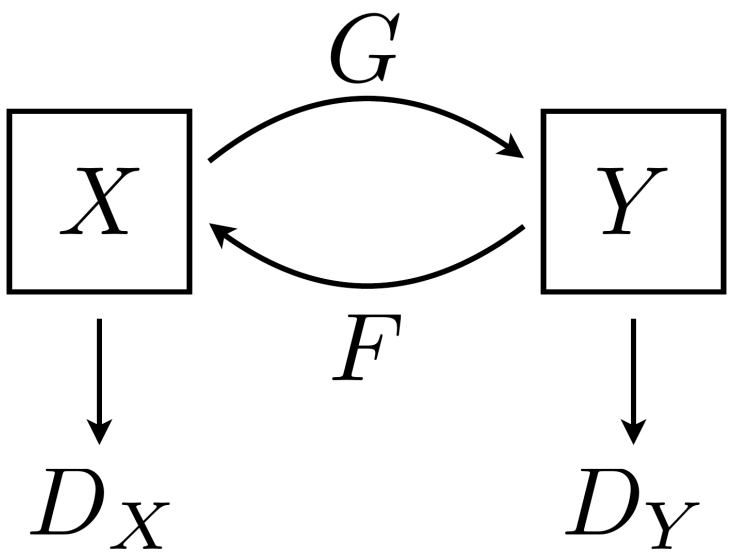
Generated sample

G_2

Generator 2 learns to map from zebra images to horse images while preserving the structure

Reconstruction

Cycle GAN



- Highly creative training strategy
- We just need two datasets, not labeling, not image aligment, or other costly information
- Generators are cycle-consistent: mapping from style 1 to style 2 and back again to original image.

Cycle GAN Applications