valeo.ai

Unsupervised learning

Alexandre Boulch

www.boulch.eu

IOGS - ATSI

Introduction

What is unsupervised learning?

Definition

Find underlying structures in unlabeled data.

Motivations

- Most of the data is unlabeled
- Annotations are expensive
- Annotations are slow

What do we do with unsupervised learning?

Dimension reduction

Keep only useful information: easy storage, computation speed up . . .

Clustering

Group data by similarity. Classification (without a guide)

Visualization

Humans do not deal well with n > 3 dimensional space.

Feature extraction

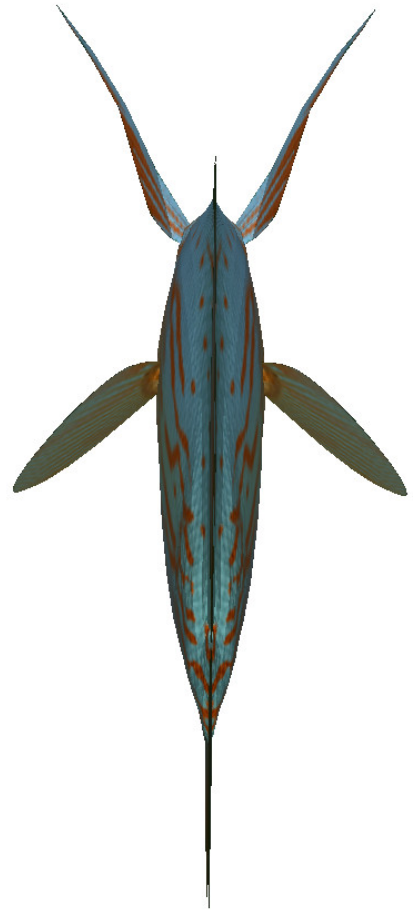
Pre-train neural network on large data (no label) to be further used on small datasets.

Dimension reduction

Principal Component Analysis

Draw a fish

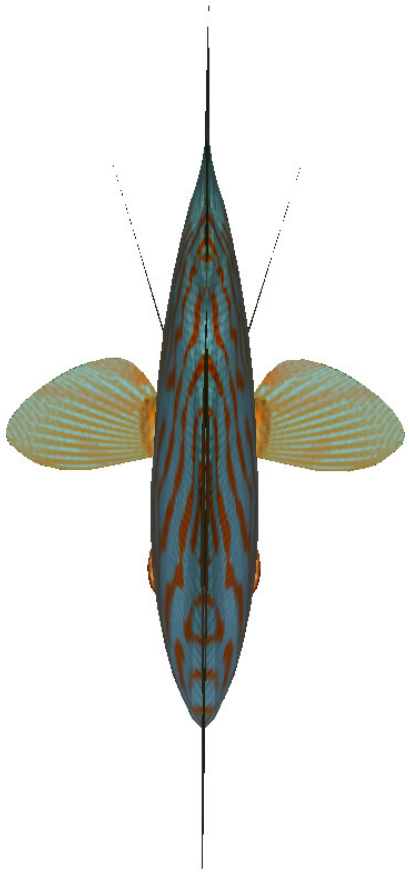
- Fishes are 3D objects
- How to create 2D representation?
- Find the best point of view
- Even better: perspective (Giotto, 1420)



Principal Component Analysis

Draw a fish

- Fishes are 3D objects



Principal Component Analysis

Draw a fish

- Fishes are 3D objects
- How to create 2D representation?

Principal Component Analysis

Draw a fish

- Fishes are 3D objects
- How to create 2D representation?
- Find the best point of view

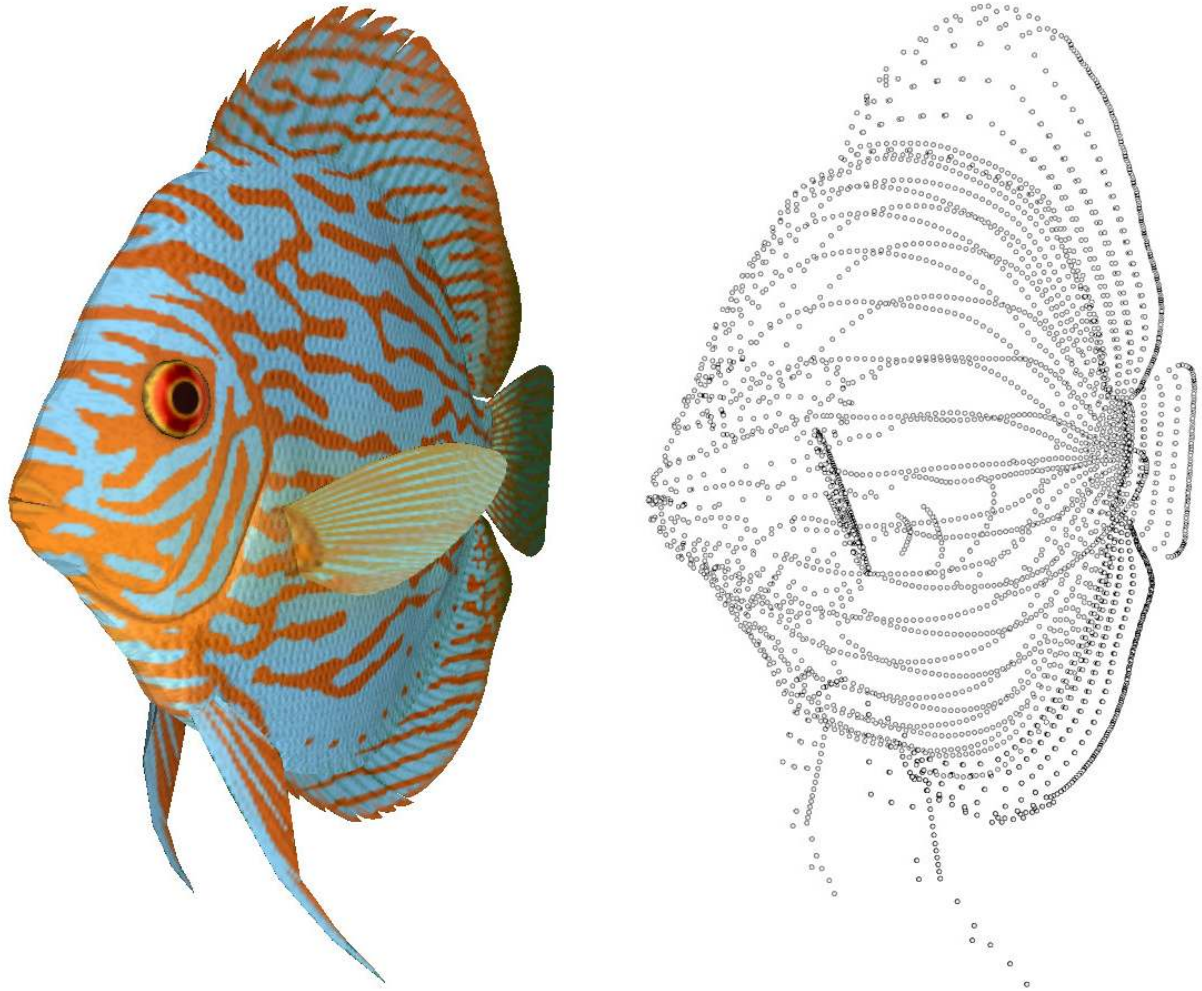
Principal Component Analysis

Draw a fish

- Fishes are 3D objects
- How to create 2D representation?
- Find the best point of view
- Even better: perspective (Giotto, 1420)

Principal Component Analysis

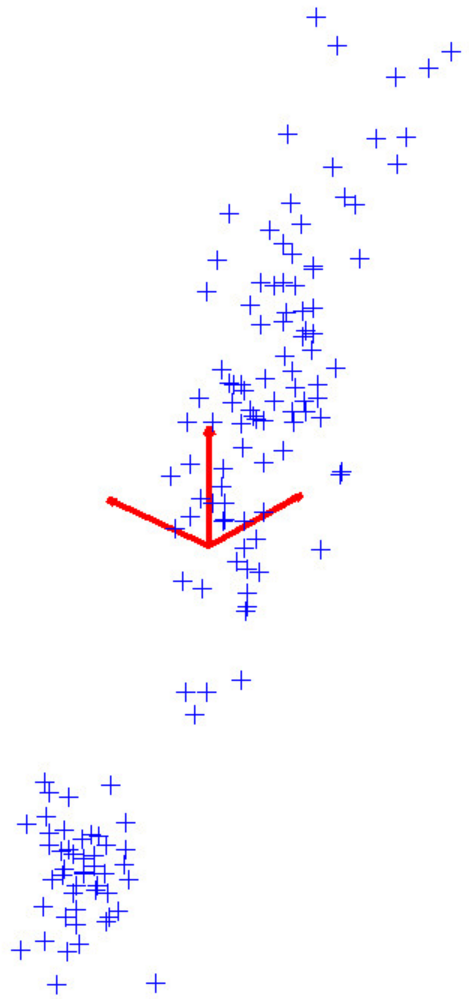
PCA is a prejection method to represent the data by dimension number reduction.



Principal Component Analysis: formalism

Linear algebra

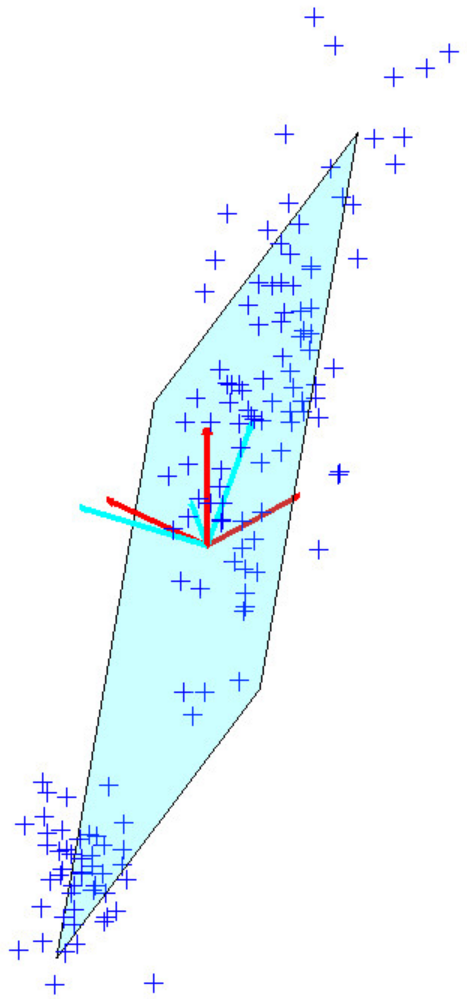
- E a vector space: structure allowing the combination of linear vectors $\mathbf{x} = (x^1, x^2, \ldots, x^n)$
- B a base: a family of *free* and *generator* vectors



Principal Component Analysis: formalism

Linear algebra

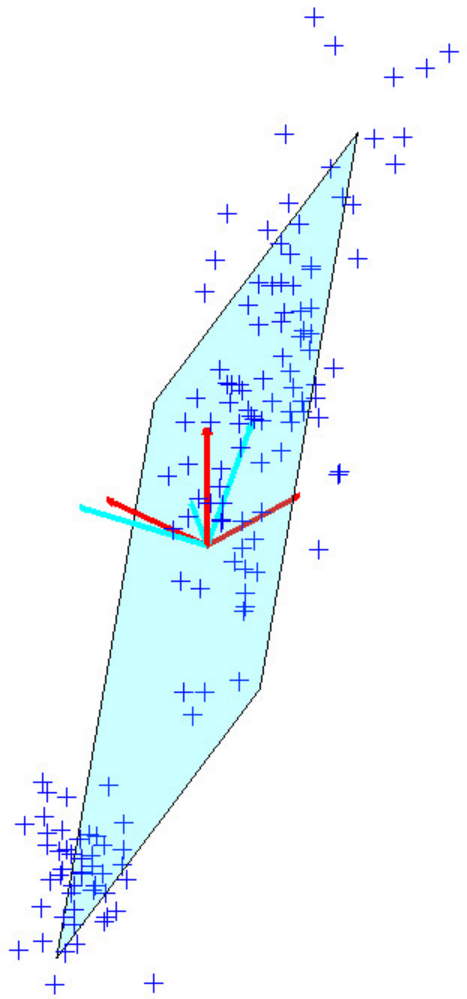
- E a vector space: structure allowing the combination of linear vectors $\mathbf{x} = (x^1, x^2, \ldots, x^n)$
- B a base: a family of *free* and *generator* vectors
- Base change: endomorphism $E \to E$, $B \to B'$



Principal Component Analysis: formalism

Linear algebra

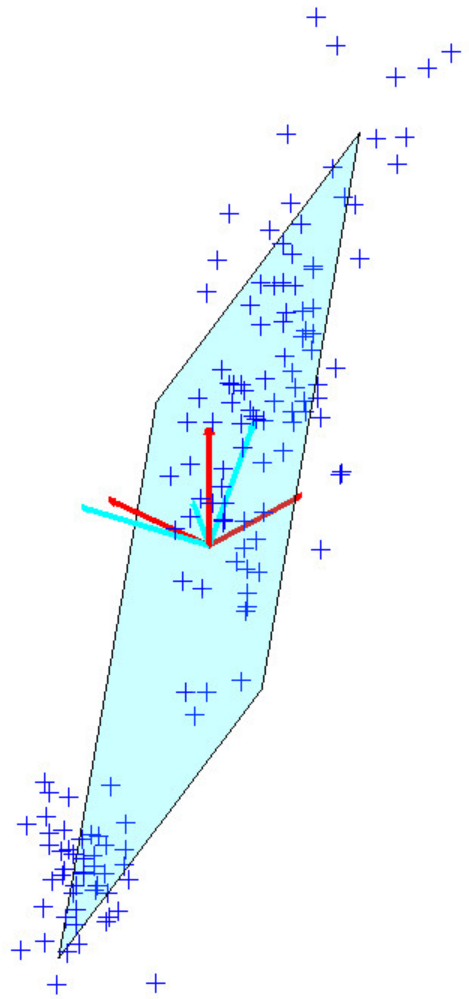
- E a vector space: structure allowing the combination of linear vectors $\mathbf{x} = (x^1, x^2, \ldots, x^n)$
- B a base: a family of *free* and *generator* vectors
- Base change: endomorphism $E \to E$, $B \to B'$
- Projection: linear application of $E \to F$, F being a sub-vector space of E



Principal Component Analysis

PCA geometric objective

PCA search for the sub vector space (with reduced dimensio) for projection which allow the more accurate projection of the data.



Principal Component Analysis: formalism

Statistics

Let X and Y be 2 Random Variables

- Average $\bar{x} = \frac{1}{N} \sum x$
- Variance $\sigma_X^2 = \frac{1}{N-1} \sum (x - \bar{x})^2$: dispersion measure
- Covariance $\sigma_{X,Y} = \frac{1}{N-1} \sum (x - \bar{x})(y - \bar{y})$: measures de correlation

Let $\mathbf{X} = (X^1, \ldots, X^n)$ a random vector:

- Variance-Covariance matrix

$$\mathrm{Var}(\mathbf{X}) = \begin{pmatrix} \sigma_{X^1}^2 & \sigma_{X^1 X^2} & \ldots & \sigma_{X^1 X^n} \\ \sigma_{X^1, X^2} & \sigma_{X^2}^2 & \ldots & \sigma_{X^2 X^n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X^1, X^n} & \sigma_{X^2, X^n} & \ldots & \sigma_{X^n}^2 \end{pmatrix}$$

Principal Component Analysis

PCA statistical objective

$$\begin{pmatrix} \sigma^2_{X^1} & \sigma_{X^1X^2} & \dots & \sigma_{X^1X^n} \\ \sigma_{X^1,X^2} & \sigma^2_{X^2} & \dots & \sigma_{X^2X^n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X^1,X^n} & \sigma_{X^2,X^n} & \dots & \sigma^2_{X^n} \end{pmatrix} \longrightarrow \begin{pmatrix} \sigma^2_{X^1} & 0 & \dots & 0 \\ O & \sigma^2_{X^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2_{X^n} \end{pmatrix}$$

PCA aims at:

- Dispersion maximization on the first dimensions of the base:
  $\sigma_{X^i} \geq 0$ and $\sigma_{X^i} \geqslant \sigma_{X^j},\ \forall i > j$
- Dimensions are not correlated: $\sigma_{X^iX^j} \to 0$

Principal Component Analysis: algorithm

Samples $\{\mathbf{x}_k = (x_k^1, \ldots, x_k^n),\ 1 \leq k \leq p\}$ of a random vector $\mathbf{X} = (X^1, \ldots, X^n)$. M matrix of $\mathbf{x}_k$s vectors.

1. Center the sample $\forall i\ X^i \to X^i - \bar{X}^i$, s.t., $B = M - \bar{M}$
2. Build the variance-covariance matrix

$$Var(\mathbf{X}) = \frac{1}{p-1}^\top BB$$

3. Diagonalize $Var(\mathbf{X})$[1]:

$$Var(\mathbf{X}) = P\Delta^\top P$$

4. Sort the eigenvalues in decreasing order (and eigenvectors)

$\Longrightarrow$ We obtain the tranfer matrix P et the eigenvalues δ_i.

symmetric, real, so can be diagonalized (Weirstrass theorem)

Principal Component Analysis: properties

- Transfer matrix $P = (\mathbf{u}^1, \ldots, \mathbf{u}^n)$ made of the vector of the new base:

$$T = PM$$

- Projection matrix $P_{1 \to l} = (\mathbf{u}^1, \ldots, \mathbf{u}^l)$ in an optimal sub-vector space:

$$T = P_{1 \to l} M$$

Principal Component Analysis: properties

- The eigenvectors $P = (\mathbf{u}^1, \ldots, \mathbf{u}^n)$ assotiated with the eigenvalues $\delta_i \propto \sigma_i^2$ sorted in increasing order
- Variance $\propto$ statistical information carried by the dimension. Link to signal theory:
  - The principal components (with a large variance) represent the signal
  - Low variance components are the noise

Principal Component Analysis: examples

Back to the fishes

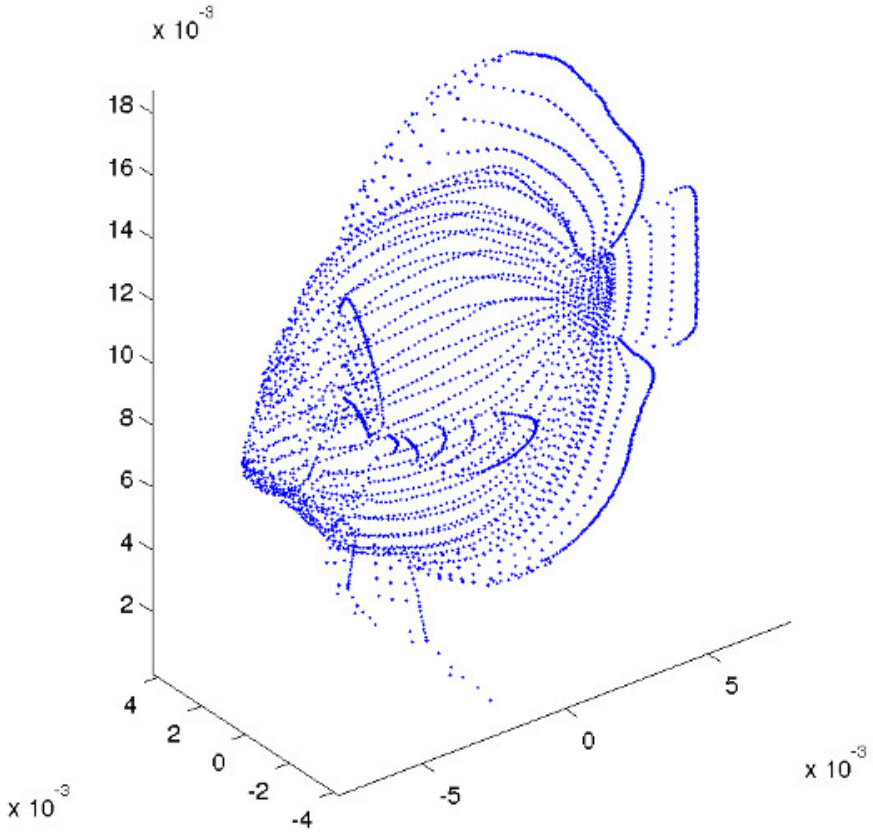
Points of $\mathbb{R}^3$ on the surface of the Discus Alenquer

Variances:

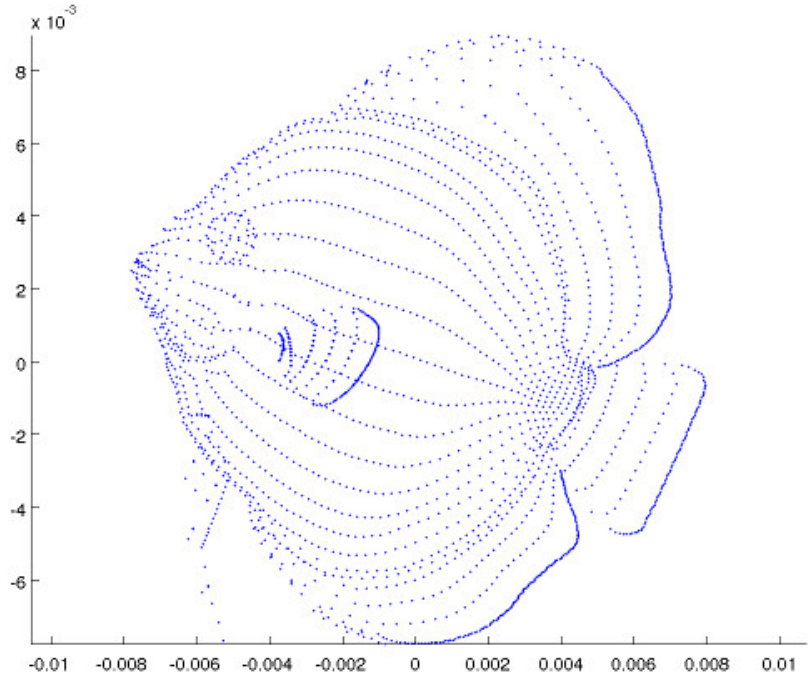
$$\Delta \propto \begin{pmatrix} 0.17 & 0 & 0 \\ 0 & 0.15 & 0.01 \\ 0 & 0 & 0.01 \end{pmatrix}$$

Eigen vector base:

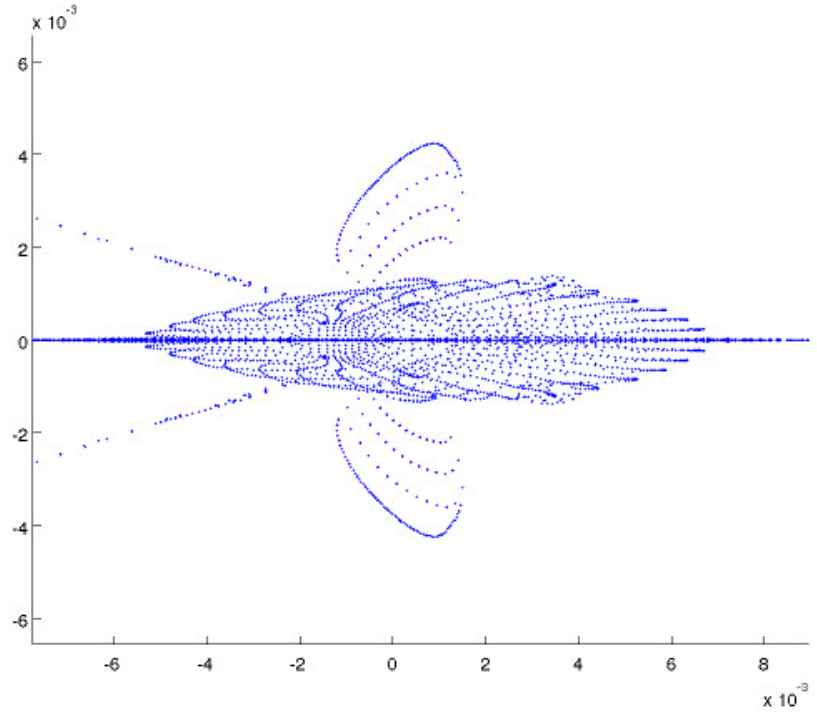
$$P = \begin{pmatrix} 0.91 & -0.42 & 0 \\ 0 & 0 & 1 \\ 0.42 & 0.91 & 0 \end{pmatrix}$$



Principal Component Analysis: examples



Back to the fishes

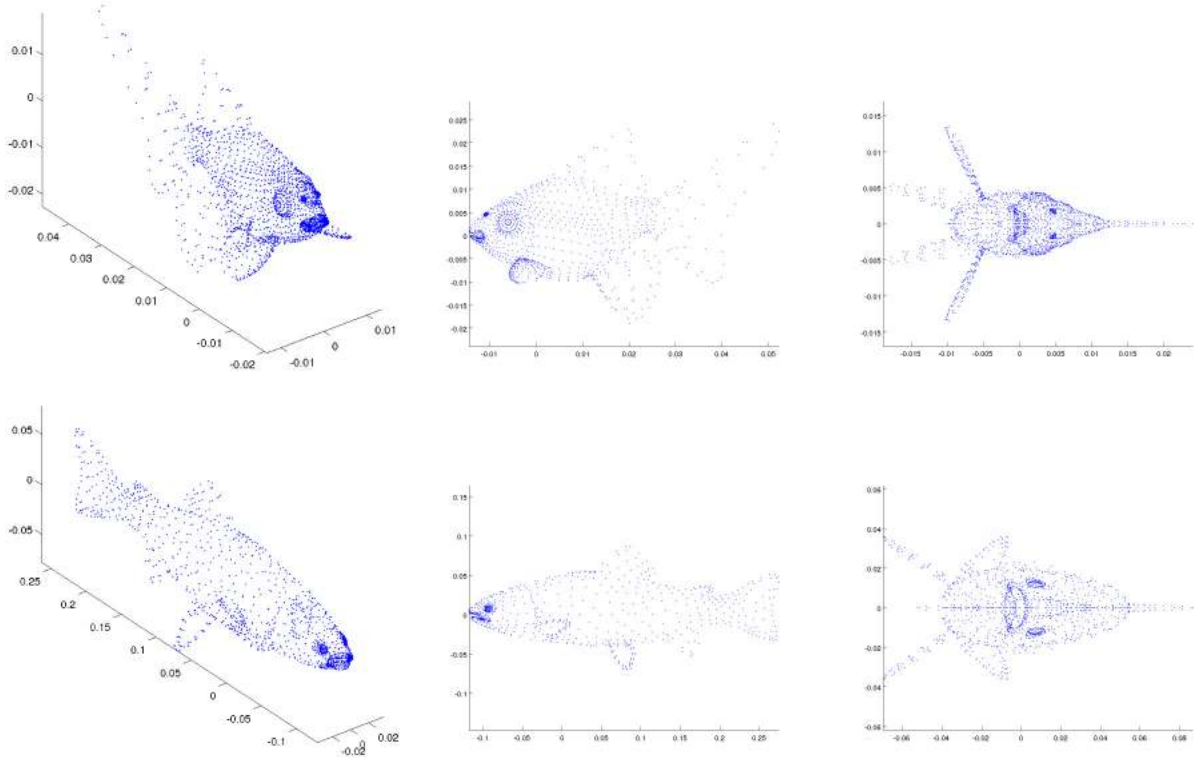


Projection on the two first components (or the two last)

Principal Component Analysis: examples

More fishes

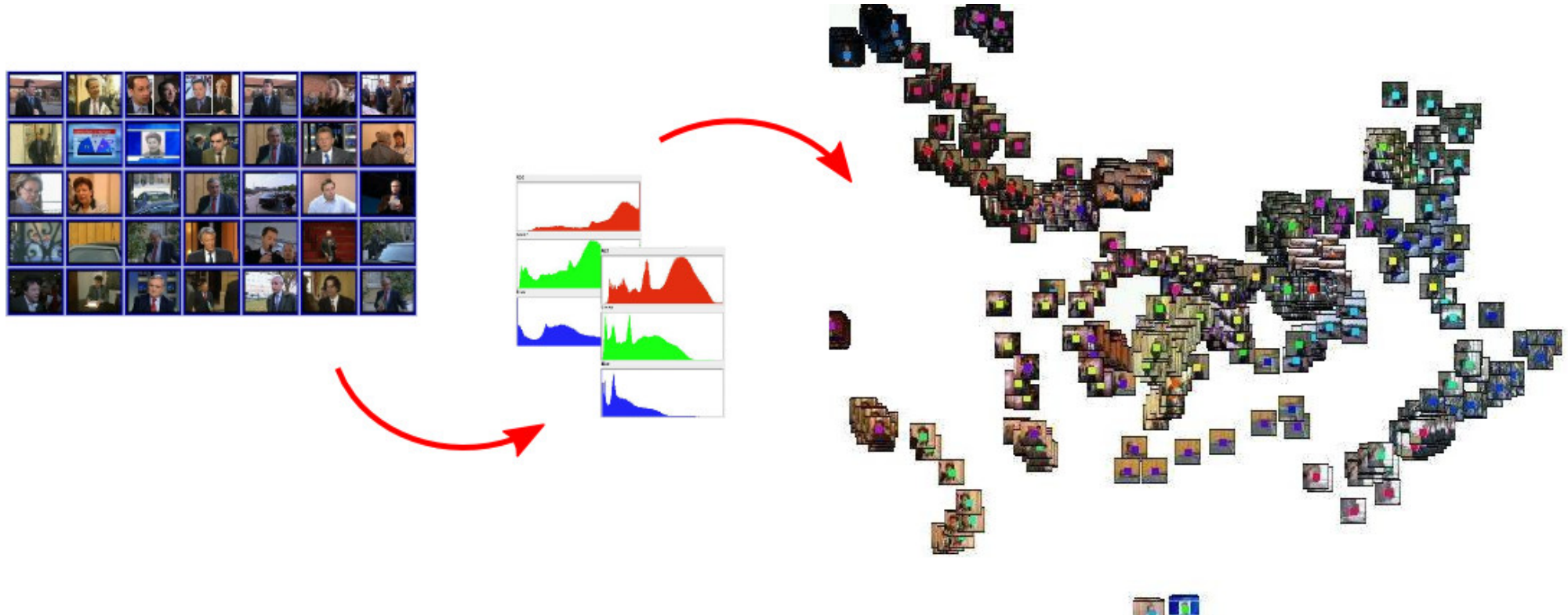
3D vs projection on the two first components (canonical representations) and the last.



Principal Component Analysis: examples

Video analysis

Video images converted to color histrograms and visualized with the 2 first components of the PCA.



Principal Component Analysis

Key points

- High dimension data representation
- Dimension reduction
- Variable decorrelation
- Based on data variance-covariance matrix diagonalization

Usage

- Data preprocessing for data analysis (see the first classes)
- Visualization

Clustering

Clustering

Definition

- Find categories for close/similar objects
- **unsupervised** classification of the unlabeled data $\{x_i | i \in \{1, \dots N\}\}$ in $\mathbb{R}^n$

Objectives

- Group similar data, requires a notion of distance
- Categorization

K-means

- Let K be the cluster number
- A cluster (indexed by j) is a group of points
- Let $u_{ji} in \{0, 1\}$ define if x_i belongs to cluster j
- Let $B = \{\beta_j | j \in \{1 \dots K\}\}$ be the cluster prototypes (caracterization of a prototype)

K-means

Algorithm minimizes:

$$J_{B,U} = \sum_{j=1}^{K} \sum_{i=1}^{N} u_{ji} \, d^2(x_i, \beta_j)$$

K-means

Algorithm

Initilize β_js, and iterate:

- Assign each x_j to the closest β_j
- Recompute the β_js according to:

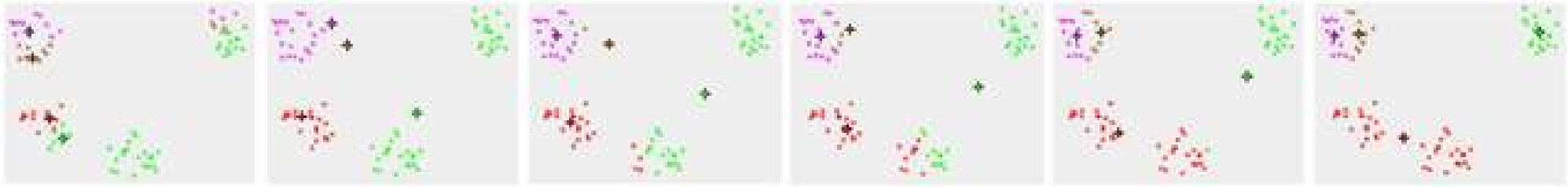
$$\beta_j = \frac{\sum_{i=1}^{N} u_{ji}\, x_i}{\sum_{i=1}^{N} u_{ji}}$$

(average location of the group)

K-means

Properties

- $J_{B,U}$ decreases at each iteration
- There is a fixed number of cluster, so the algorithm **converges**
- **But** the solution may not be optimal (local minimum)



Initialization

- **Initialization** is important
- E.g., chose the initial β_j among the data x_i

K-means

Statistical variant

- Parameters of *Gaussian Mixture Model* (GMM) estimated by the *Expectation-Maximization* algorithm.
- x_i is the realisation of a random vector, modeled with a Gaussian mixture:

$$p(x_i) = \sum_{1}^{K} p(x_i|k)P(K)$$

- $d(x_i, \mu_k) \rightarrow p(x_i|k) \propto \exp(-||x_i - \mu_k||^2/2\sigma_k^2)$
- Estimate: $\forall k, P(k), \mu_k, \sigma_k$

K-means

Variants and tricks

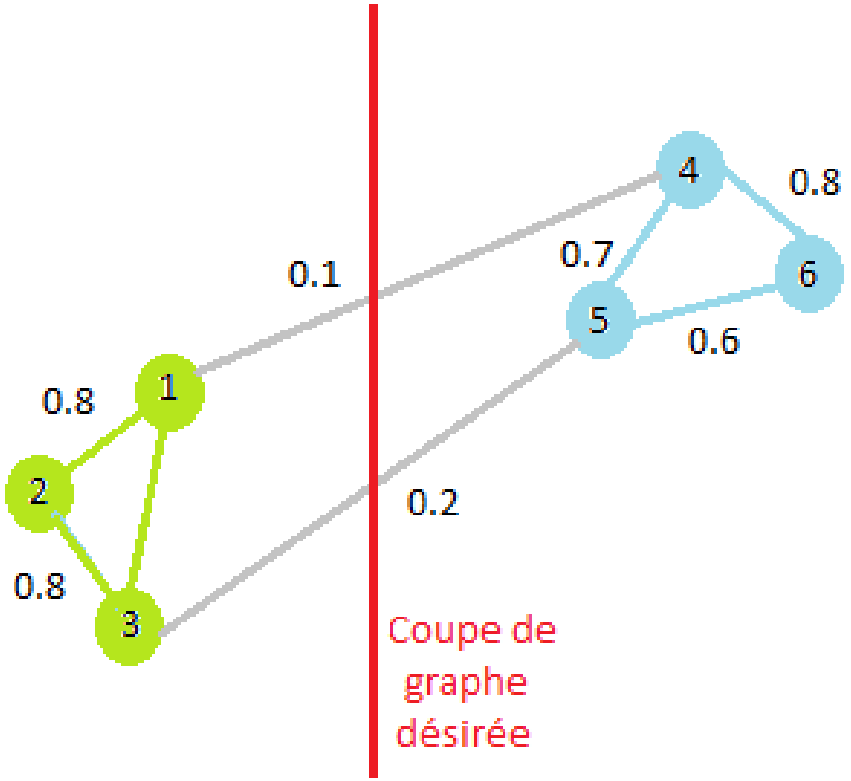
- Fuzzy C-means: $u_{ji} \in [0, 1]$, a point can belong to several clusters
- Different distance: Mahalanobis distance (FCM), complete covariance matrix (GMM)
- Outliers: if $\forall k d(x_i, \mu_k)$ to high, $x_i \rightarrow \text{Noise category}.$
- Criteria for K estimation
- μ_k estimation: uniformly spread among the data

Clustering: other approaches

Partitionnement spectral:

- Similarity matrix,
- Dimension reduction (first eigen vectors)
- K-means

$\Longrightarrow$ complex objects, non vectors



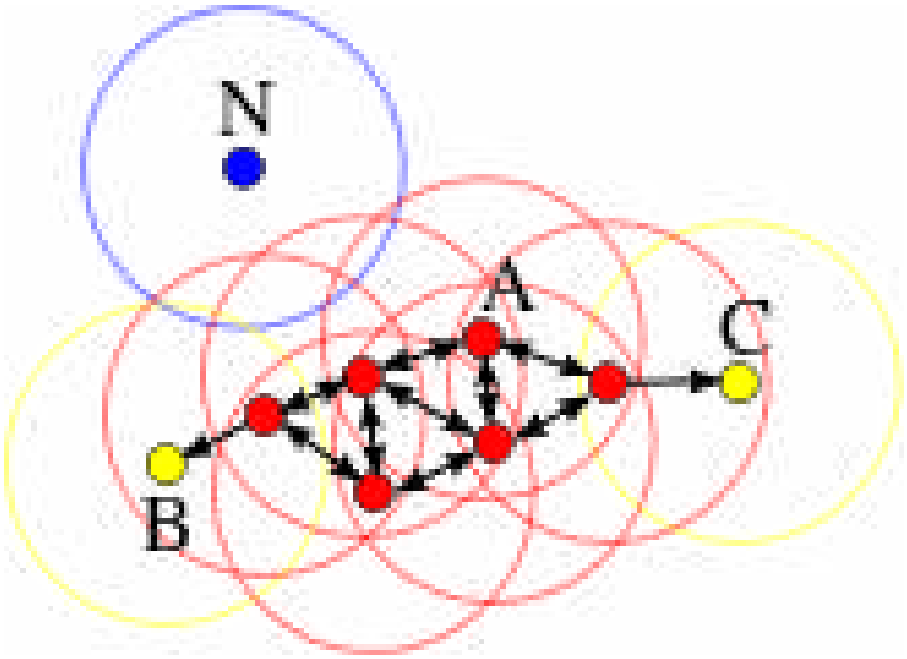
https://fr.wikipedia.org/wiki/Partitionnement_spectral

Clustering: other approaches

DBSCAN

- Data partionning in categories of MinPts points in a radiusϵ
- Going through the data step by step to add in a category

$\Longrightarrow$ Automatic estimation of the number of categories,
$\Longrightarrow$ deal with outliers



https://fr.wikipedia.org/wiki/DBSCAN

Self-supervised learning

Overview

- Principles
- Image-based transformations
- Contrastive approaches
- SimCLR

Self-supervised learning

Objective

Create a good **features** for a **downstream task**.

Pre-training of the network

Create a **pretext task** to train the network on.
The labels for task a generated automatically.

Downstream task

This the real final objective. It could be classification, regression...
It is trained in a supervised manner.

Image-based transformation

Key idea

To predict the transformation of an image, you must \textit{understand} what is in the image.

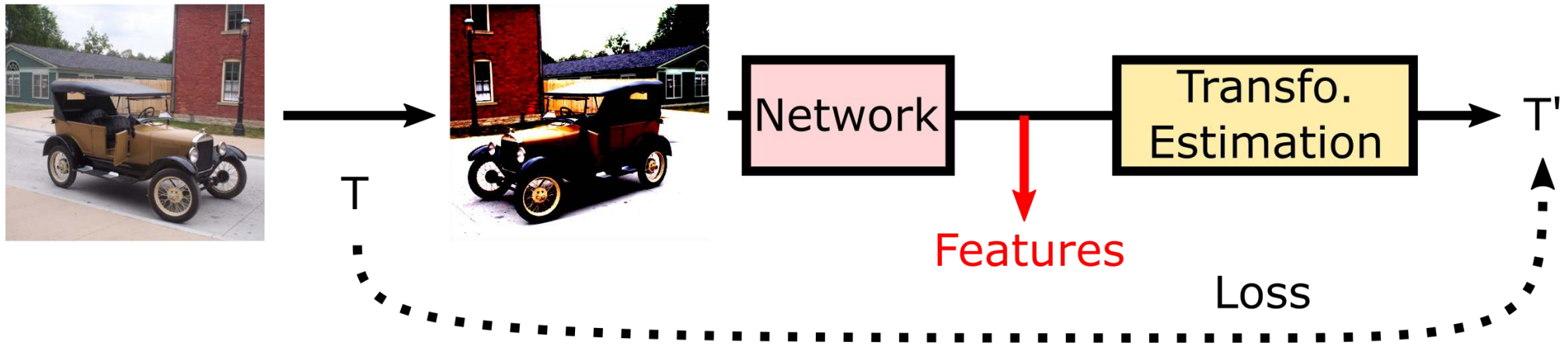


Image-based transformation: rotation

Transformation

Random rotation of the image.

Four classes: $0°, 90°, 180°, 270°$.

Simple classification problem.

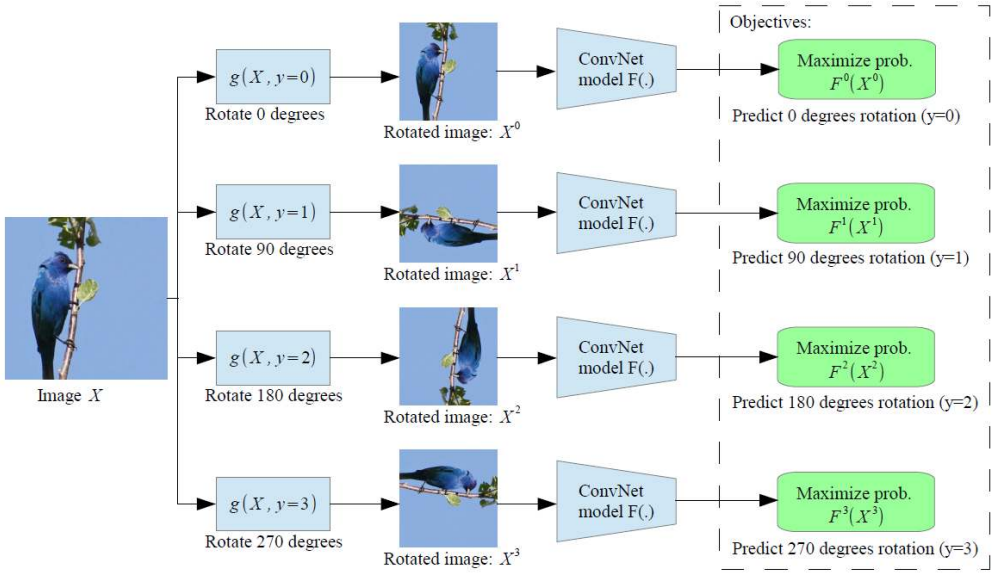


Image-based transformation: rotation

Transformation

Random rotation of the image.

Four classes: $0°, 90°, 180°, 270°$.

Simple classification problem.

Semi-supervised learning

Pre-training: all data (no label)

Target: part of the data with labels

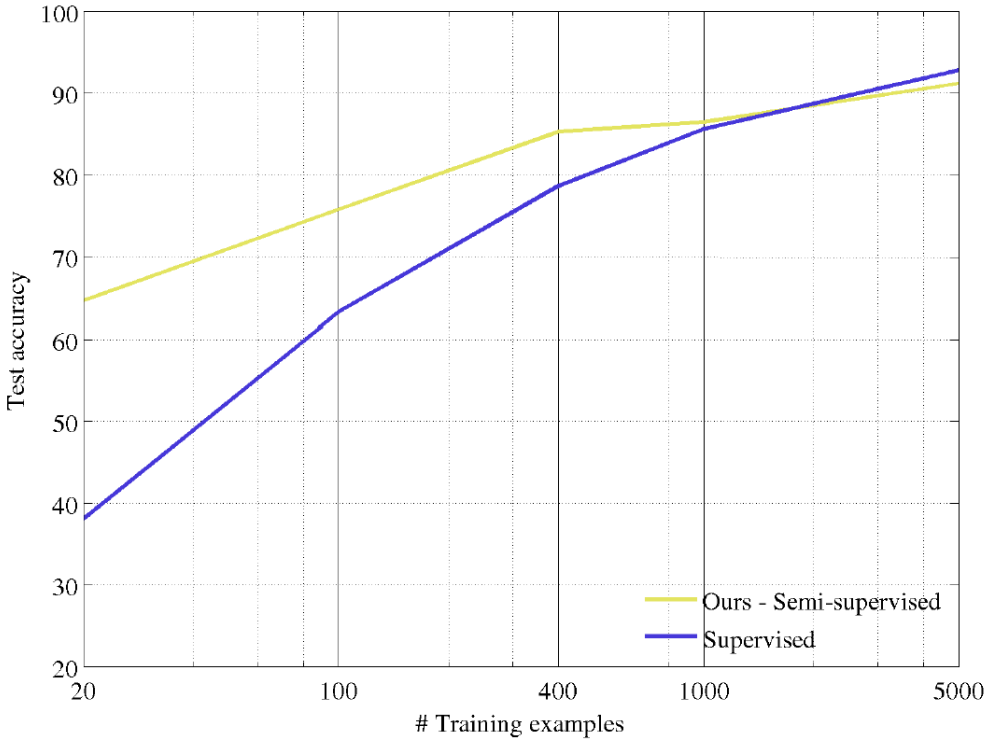


Image-based transformation: rotation

Transformation

Random rotation of the image.

Four classes: $0°, 90°, 180°, 270°$.

Simple classification problem.

Semi-supervised learning

Pre-training: all data (no label)

Target: part of the data with labels

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Image-based transformation: relative position

Transformation

Create a pair of patch, find their relative position.
To solve problem, you need to understand the object.

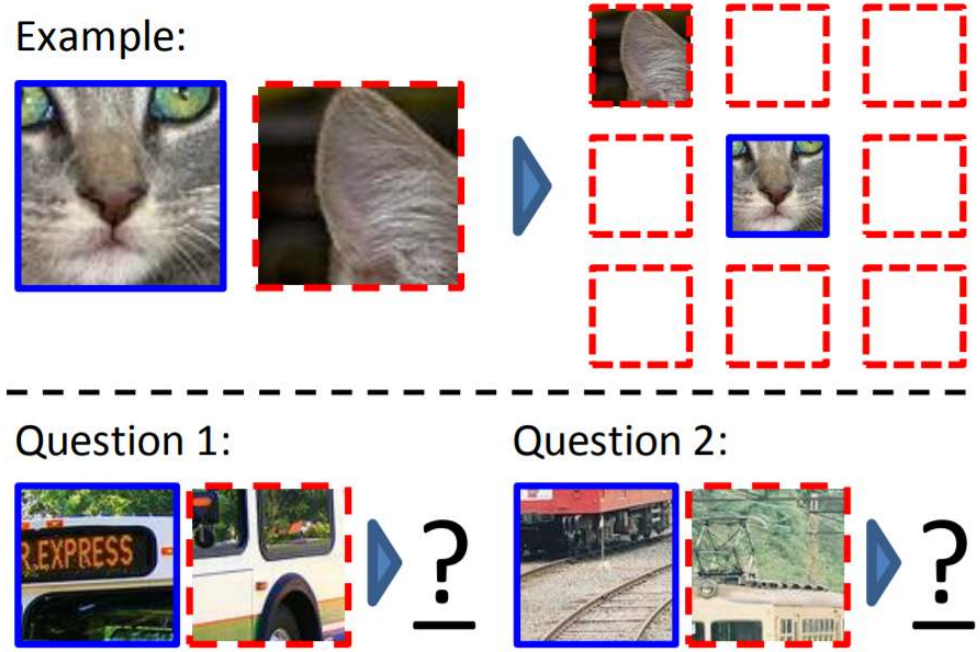


Image-based transformation: relative position

Transformation

Create a pair of patch, find their relative position.
To solve problem, you need to understand the object.

Problem

Classification with 8 classes

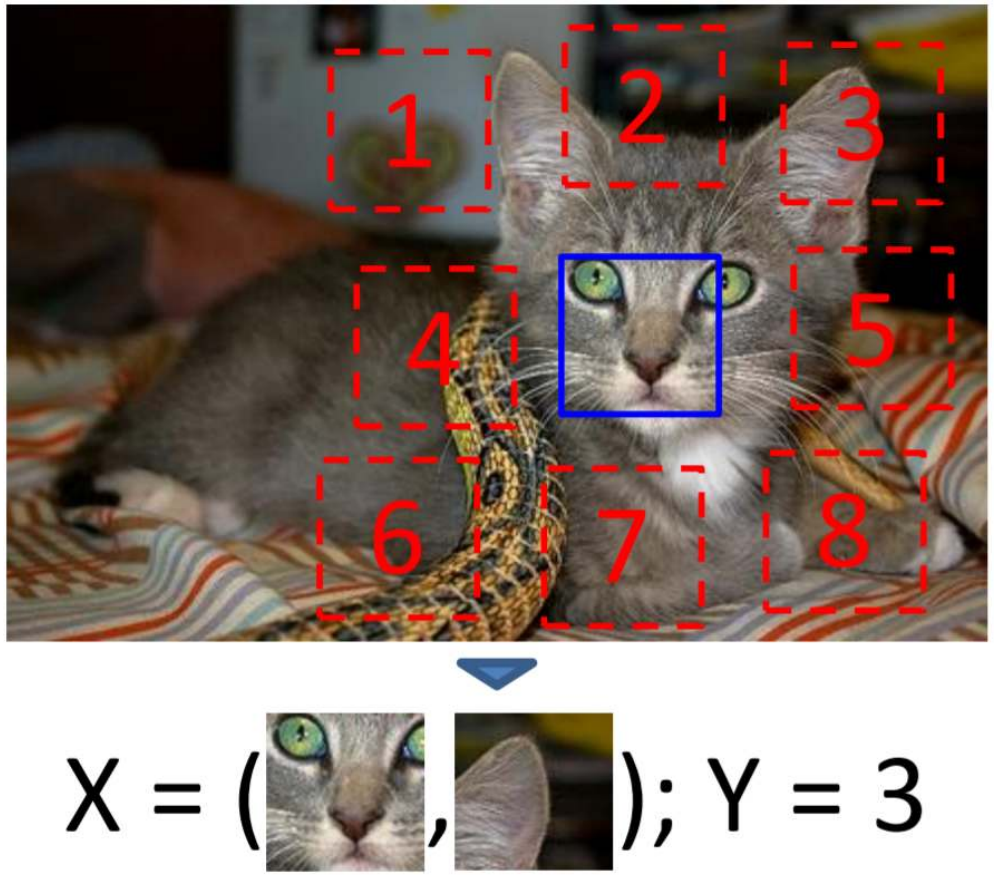


Image-based transformation: relative position

Transformation

Create a pair of patch, find their relative position.
To solve problem, you need to understand the object.

Problem

Classification with 8 classes

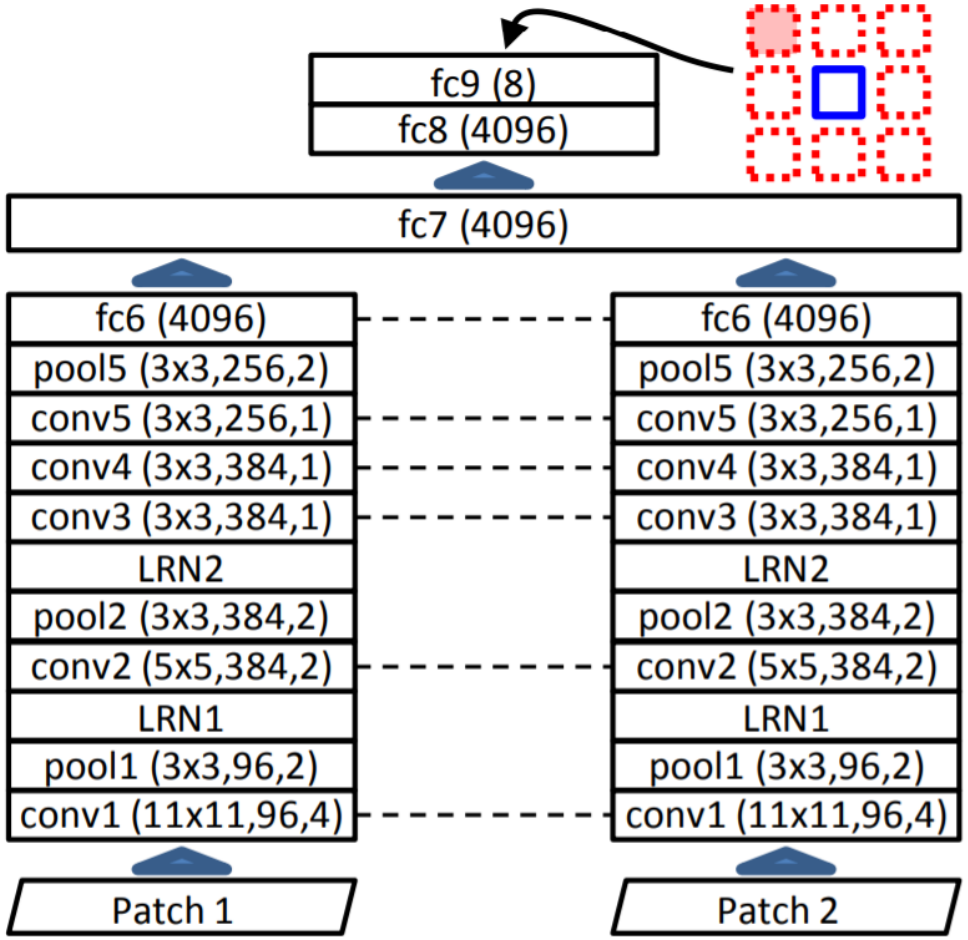
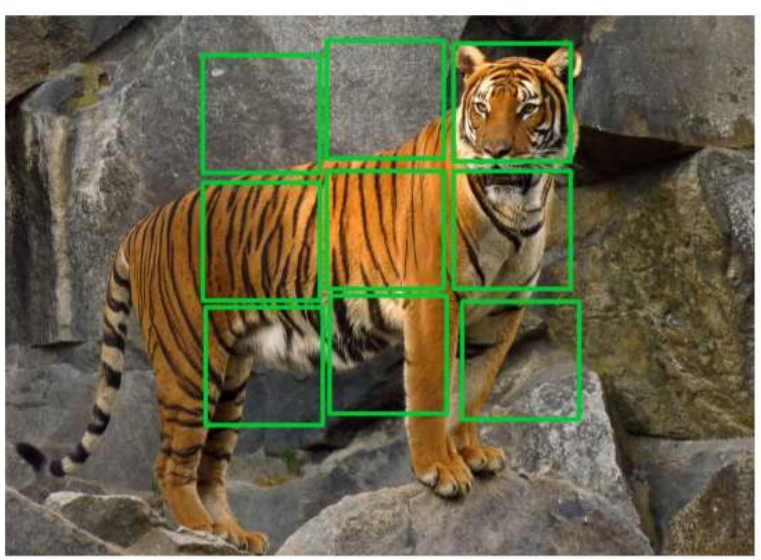


Image-based transformation: jigsaw puzzle}

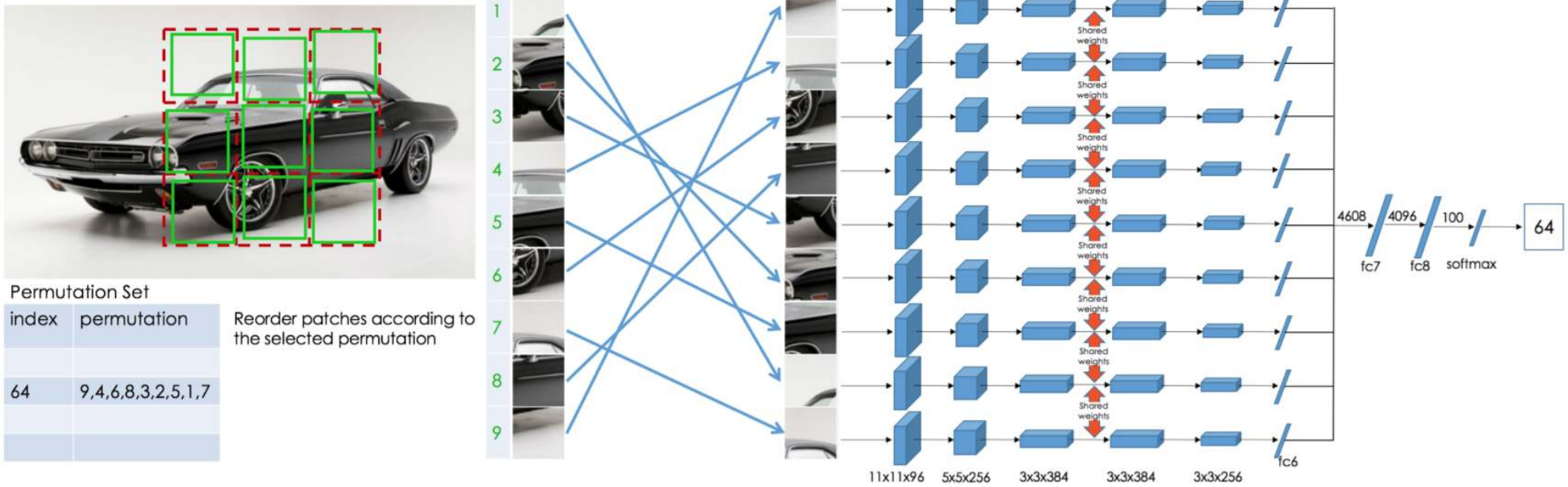


(a)

(b)

(c)

Image-based transformation: jigsaw puzzle}



Contrastive methods

Image based methods

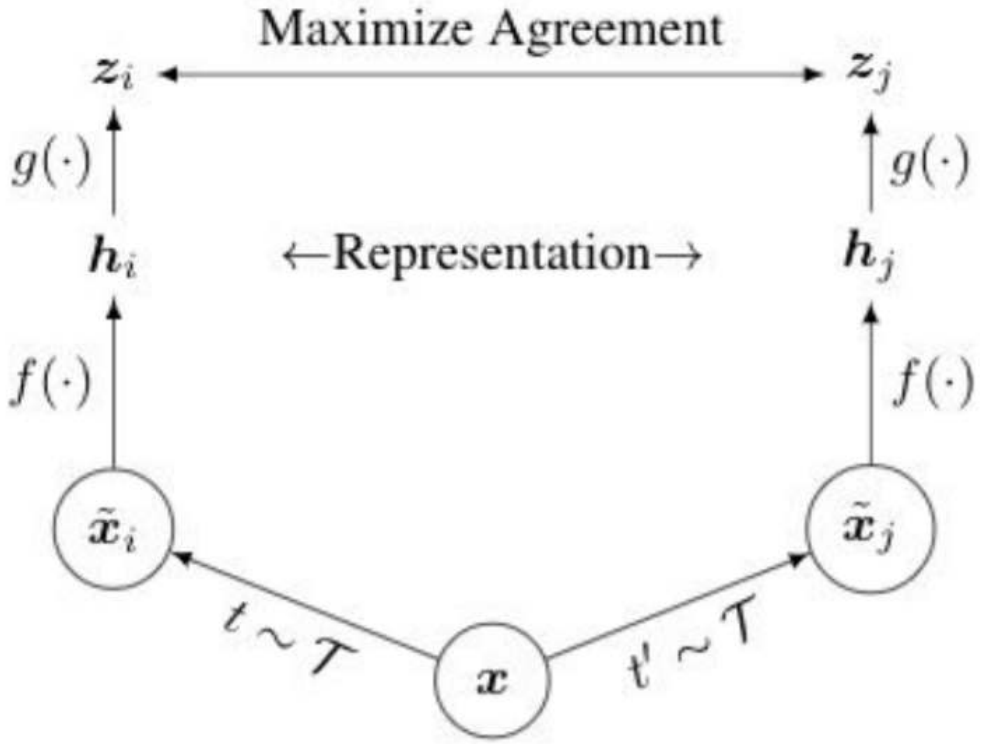
Predict a transformation of an image.

but may not require a complete knowledge of the object.

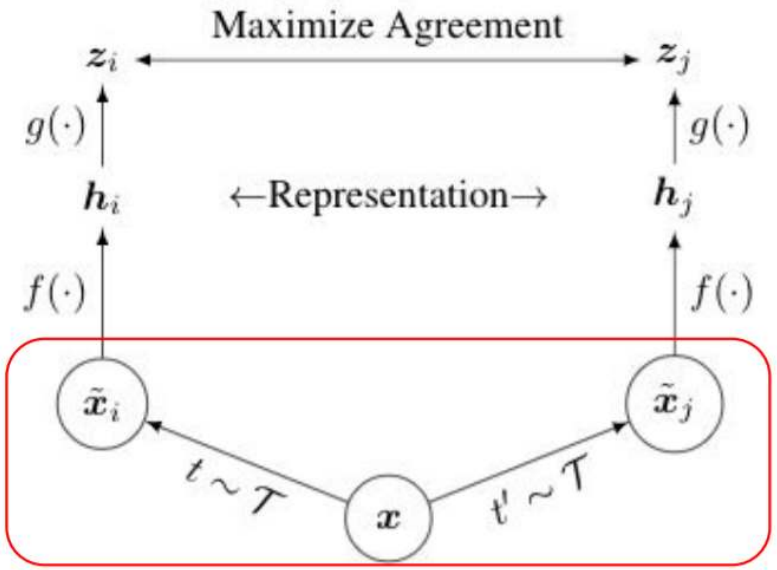
What properties the pre-trained network should have ?

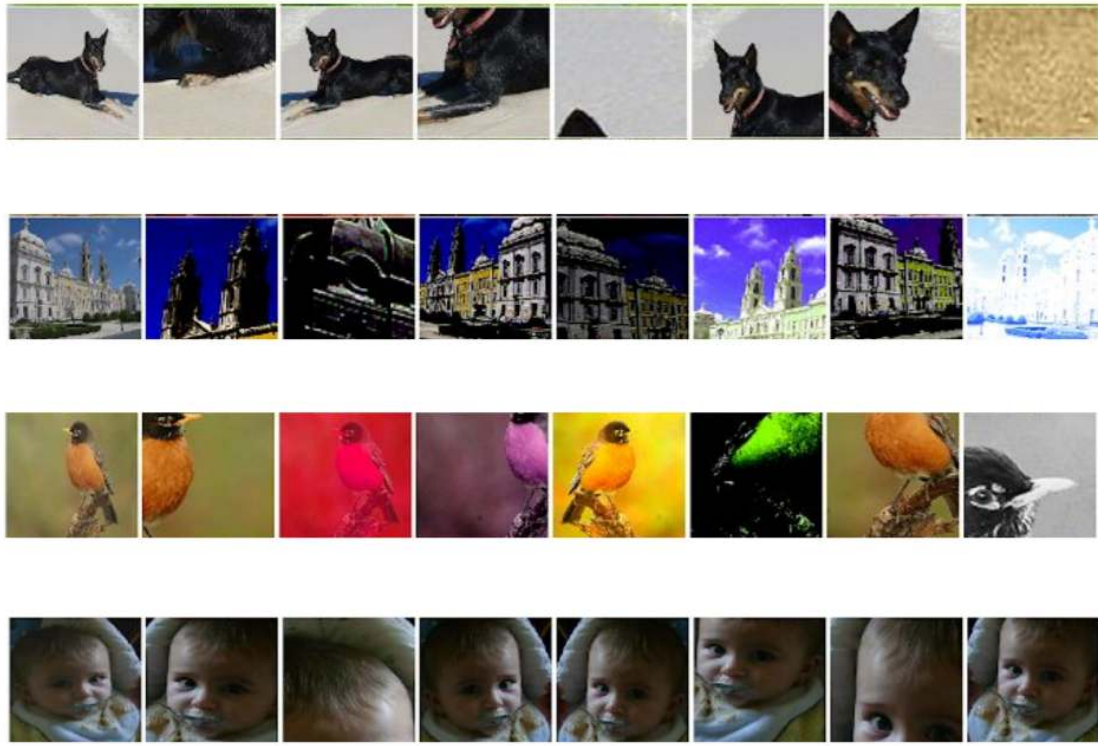
- robust to image variation (illumination, deformation)
  $\rightarrow$ Produce identical features for the same object
- disciminative with respect to different objects
  $\rightarrow$ Produce different features for different objects

SimCLR

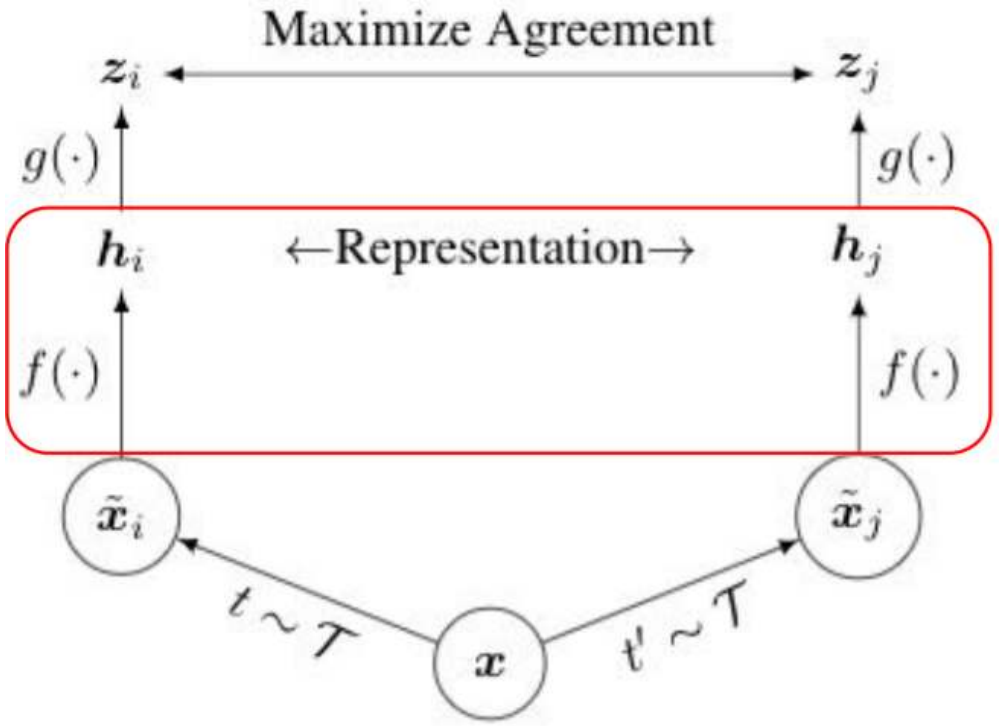

Maximize Agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
←Representation→
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
x
$t' \sim \mathcal{T}$

SimCLR

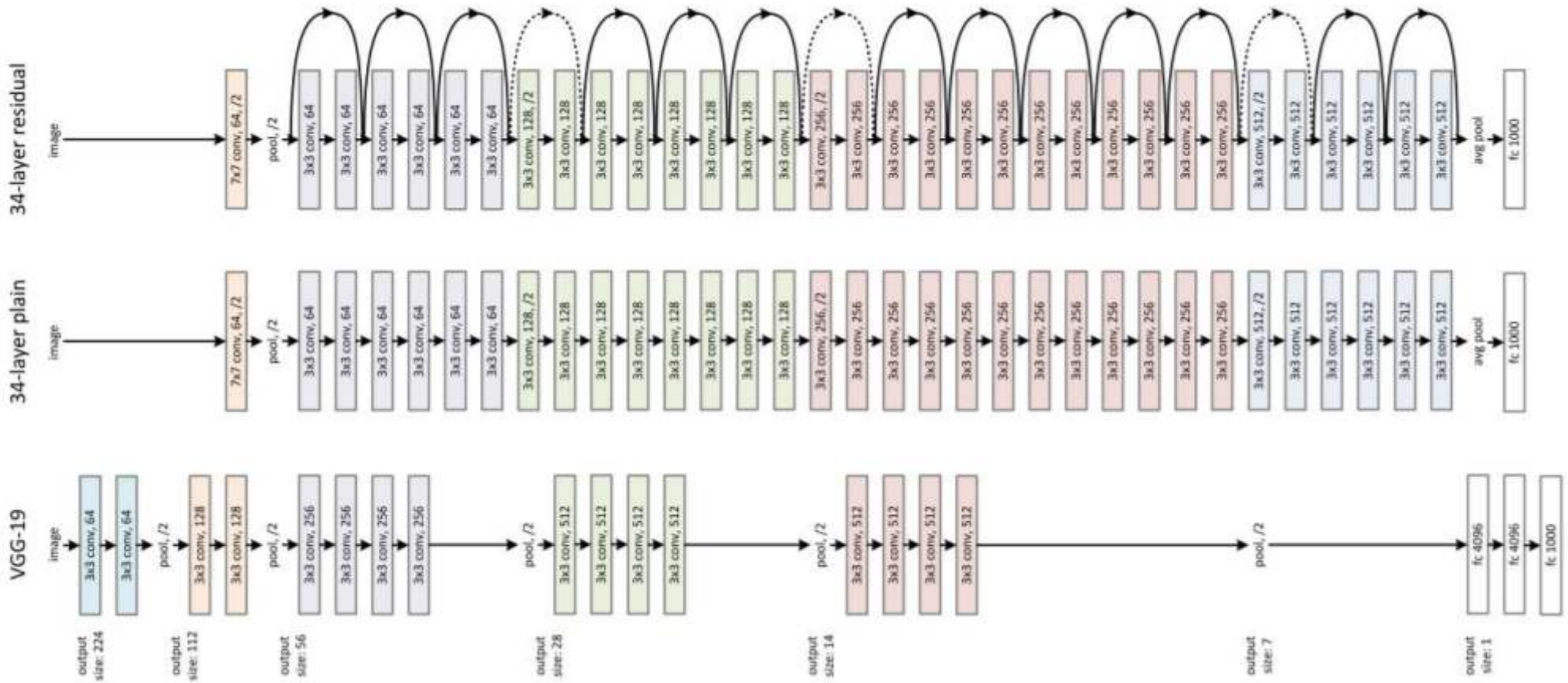

Maximize Agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
←Representation→
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
x
$t' \sim \mathcal{T}$



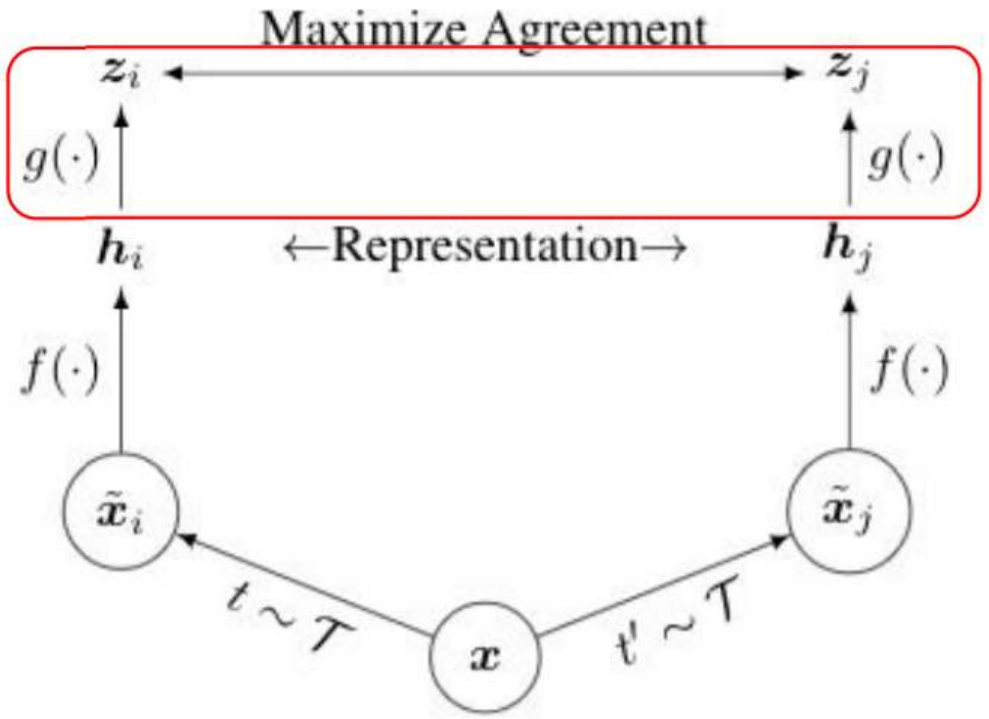
SimCLR


Maximize Agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
←Representation→
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
x
$t' \sim \mathcal{T}$

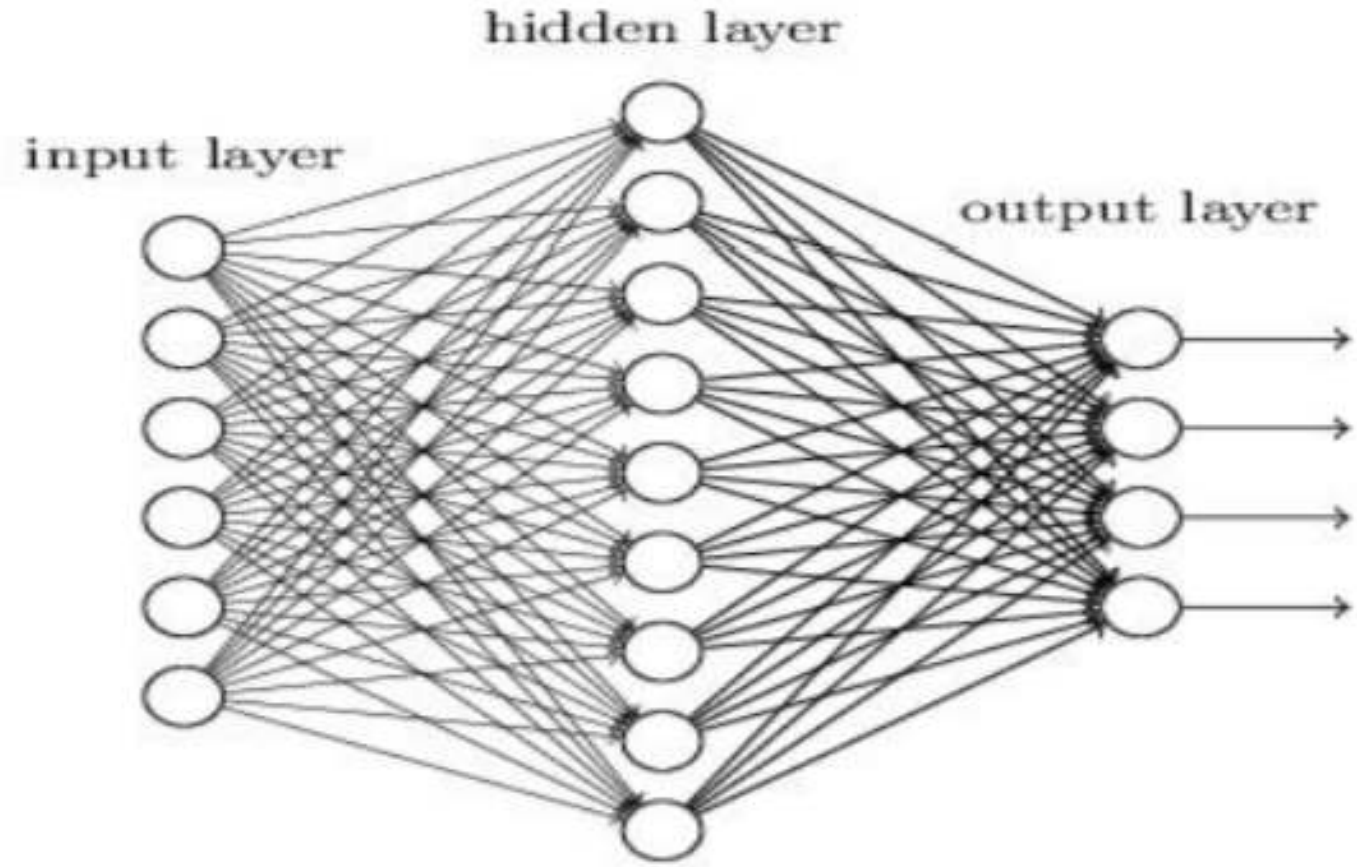
SimCLR


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128
3x3 conv, 256
3x3 conv, 512
avg pool
fc 1000
34-layer plain
VGG-19
fc 4096
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1

SimCLR


Maximize Agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
←Representation→
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
x
$t' \sim \mathcal{T}$

SimCL



Loss function

Let

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{||\mathbf{u}^\top \mathbf{v}||_2}$$

The loss function is:

$$\ell_{i,j} = -\log \frac{\exp(sim(\mathbf{z}_i \mathbf{z}_j))/\tau}{\sum_{k=1}^{2N} \mathbb{I}_{k \neq i} exp(sim(\mathbf{z}_i \mathbf{z}_k))/\tau}$$

Analysis

A **cross entropy** applied label corresponding to the pair generated from x_i.

Conclusion

Unsupervised learning is one of the big thing in machine learning now.

- Can we extract better/general features?
- Can we reduce the training time?
- Do we really exploit all the information in the data?