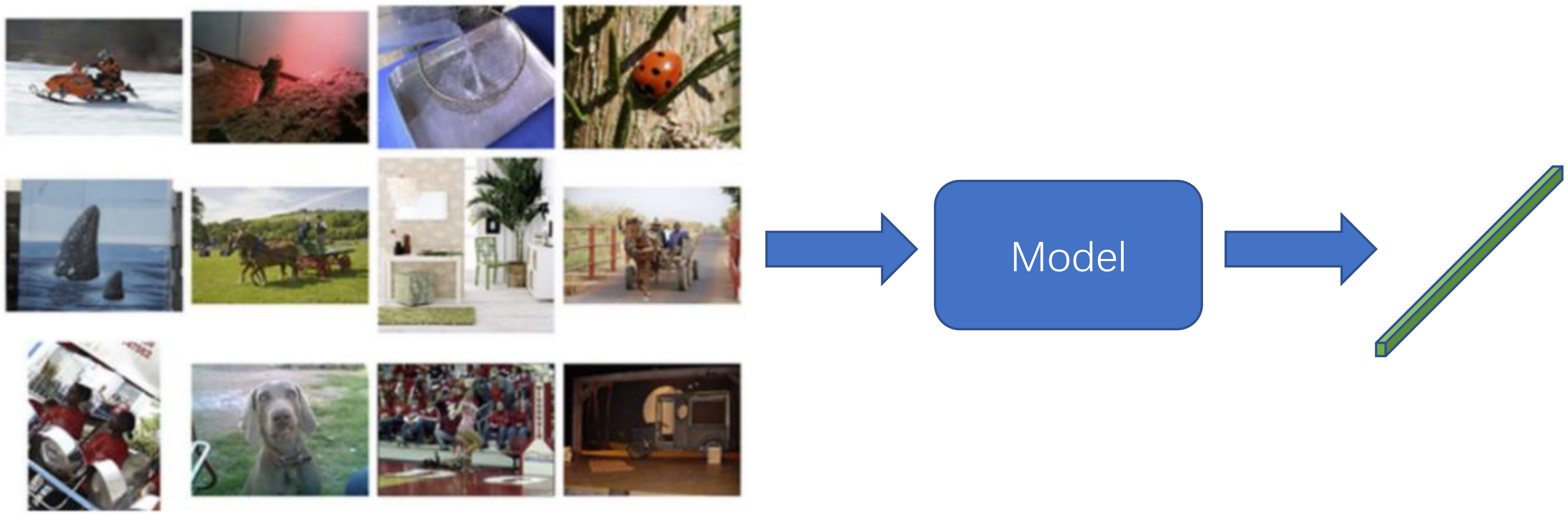
Convolutional Neural Networks – Part 4 Introduction to Representation Learning

ECE 449

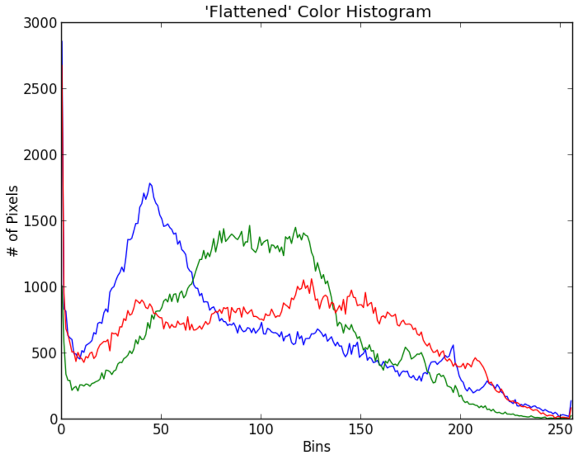
Representation Learning

- Learn the representation/feature/embedding of the data
  - The learned semantic information of the representation depends on specific tasks.
  - The “model” has several names: backbone, encoder, embedding network, feature extractor, etc.

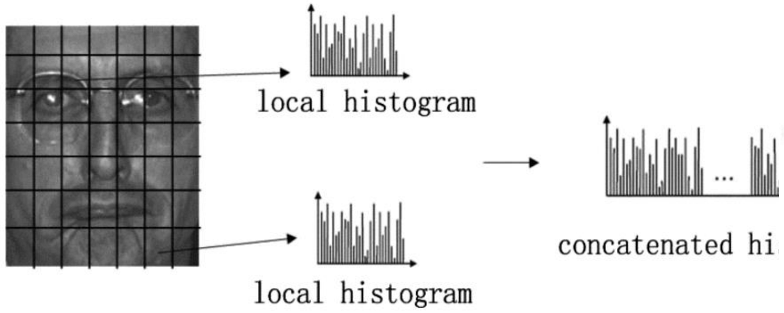


Before Deep Learning

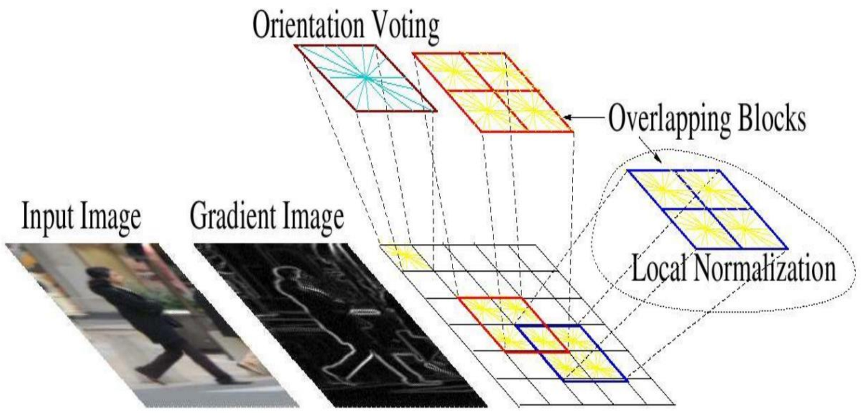
- Use feature engineering instead of representation learning
  - Hand-crafted, not learned
  - Not task specific



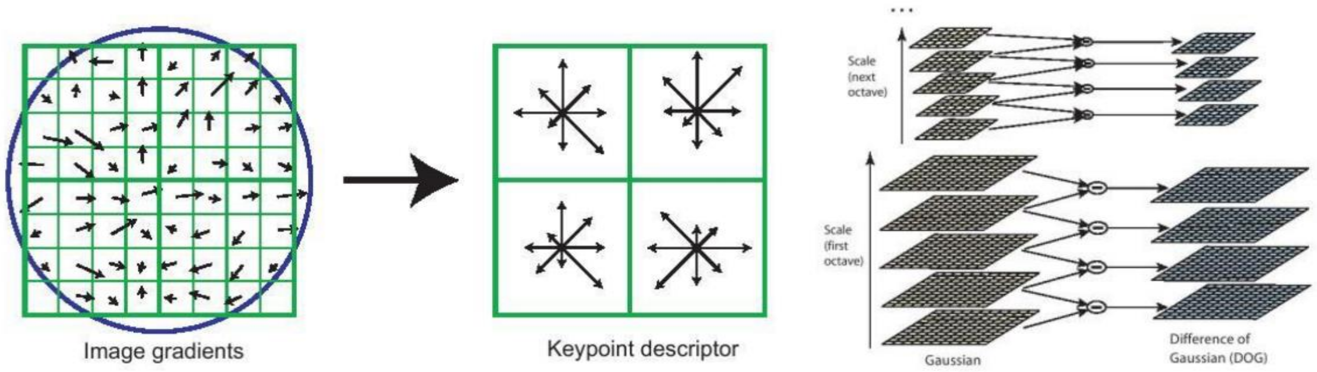
Color Histogram



LBP



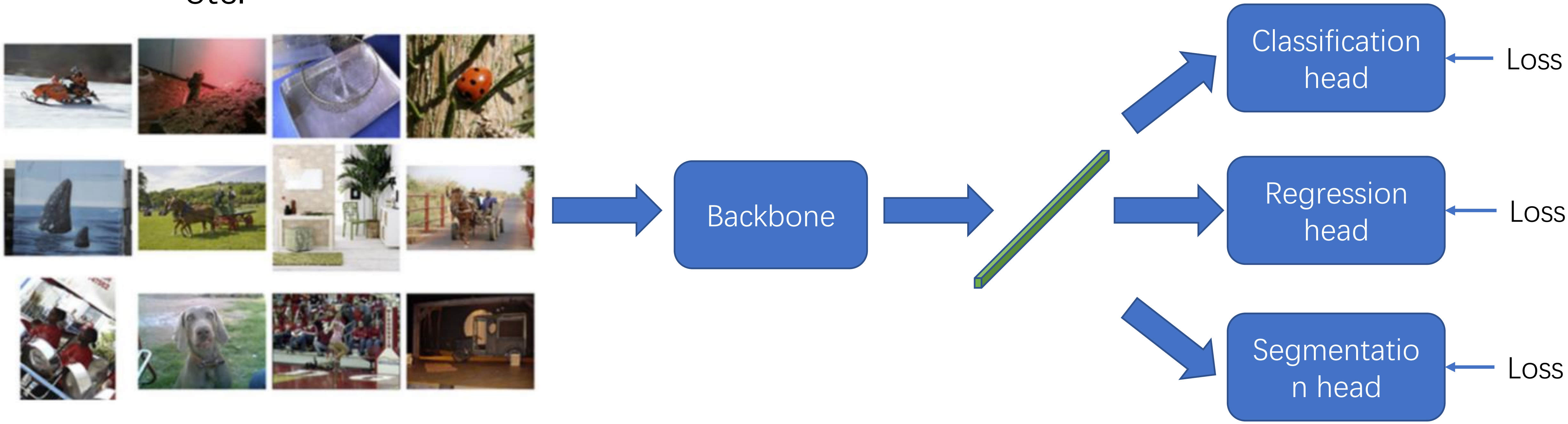
HOG



SIFT

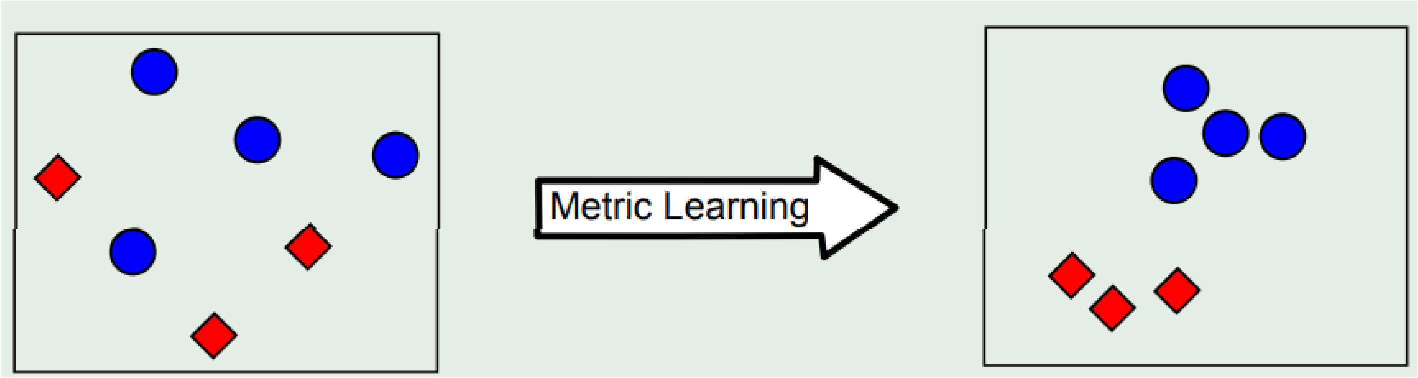
How to Learn Representations?

- Treat samples individually with task specific head
  - Classification head, regression head, segmentation head, projection head, etc.



How to Learn Representations?

- Learn relations among training samples
  - Distance metric learning, reduce intra-class distances and enlarge inter-class distances



Learn Representations

- Learn representation with individual samples
- Distance metric learning
  - Distance metrics
  - Contrastive learning and ranking losses
- Relation to other tasks
  - Self-supervised learning
  - Transfer learning
  - Multi-task learning

Learn Representation with Individual Samples

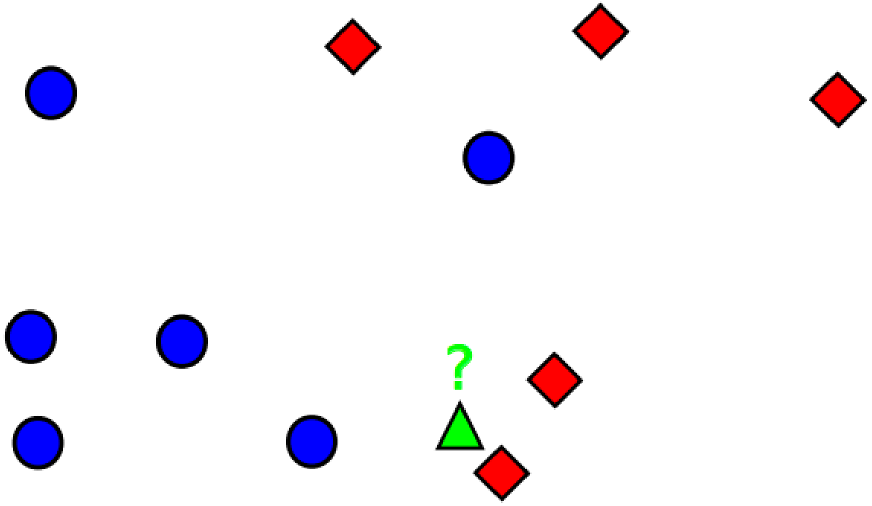
- Given the task specific labels $\boldsymbol{y}$ (for supervised tasks), backbone $f_{\boldsymbol{\theta}}$, task specific head $g_{\boldsymbol{\varphi}}$, and designed loss function l, we aim to learn $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$,

$$\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}} = \arg\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} l\left(g_{\boldsymbol{\varphi}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})\right), \boldsymbol{y}\right)$$

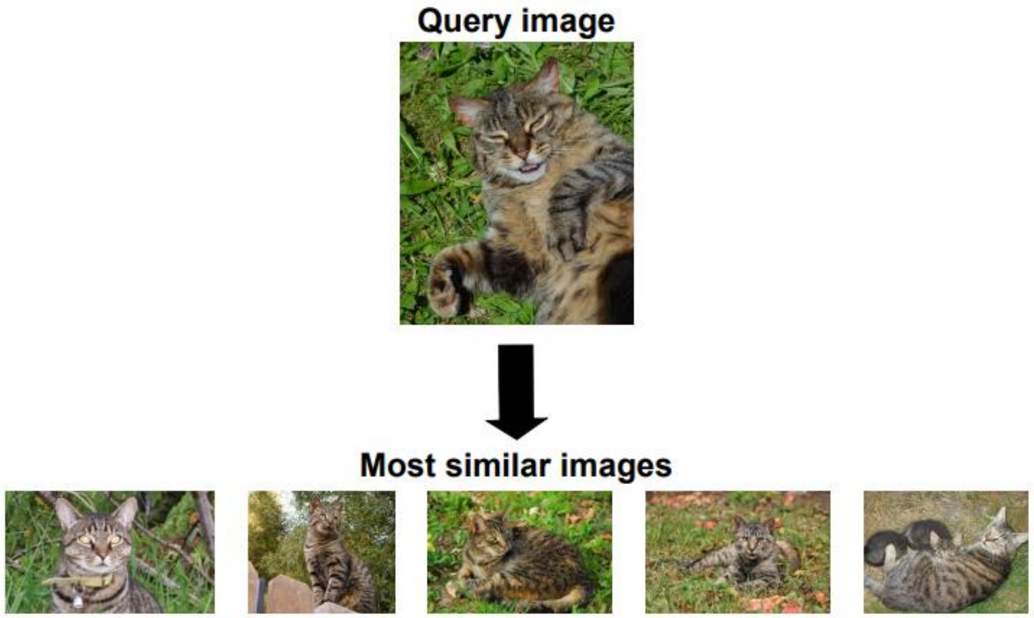
- The learned representations $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be used in downstream tasks
- Example: face recognition and verification

Distance Metric Learning

- Distance Metric learning is to learn a distance metric for the input space of data from a given collection of pair of similar/dissimilar points that preserves the distance relation among the training data pairs.



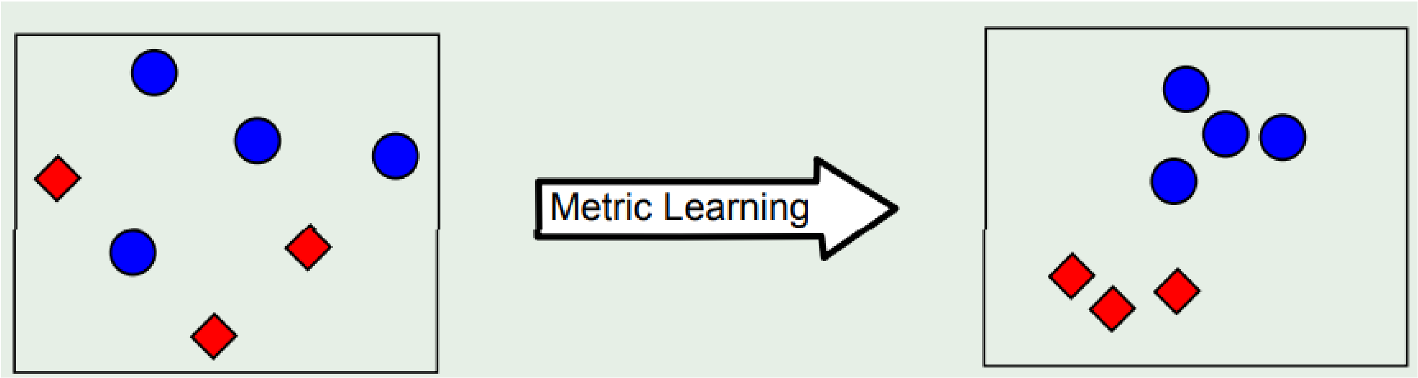
Example 1



Example 2

Metric Learning

- Adapt the metric to the problem of interest.
- The notion of good metric is problem-dependent
  - Each problem has its own semantic notion of similarity, which is often badly captured by standard metrics (e.g., Euclidean distance).
- Solution: learn the metric from data
  - Basic idea: learn a metric that assigns small (resp. large) distance to pairs of examples that are semantically similar (resp. dissimilar).



Distance Metrics

- Distance function
  - A distance over a set X is a pairwise function $d : X \times X \rightarrow R$ which satisfies the following properties $\forall x, x', x'' \in X$:
    - $d(x, x') \geqslant 0$ (non-negativity)
    - $d(x, x') = 0$ if and only if $x = x'$ (identity of indiscernibles)
    - $d(x, x') = d(x', x)$ (symmetry)
    - $d(x, x'') \leqslant d(x, x') + d(x', x'')$ (triangle inequality)
- Example:
  - Euclidean distance, $||x-x'||$
  - KL divergence, $D(p||q)=\sum p*\log(p/q)$, distance?

Minkowski Distances

- Minkowski Distance is a generalization of Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
  - A common example of this for binary vectors is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $r = 2$. Euclidean distance
- $r \to \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors
  - $d(x, y) = \max_i |x_i - y_i|$.

The Mahalanobis Distance

- Definition

$$d(\mathbf{x}, \mathbf{y}) = \ ((\mathbf{x} - \mathbf{y})^T \boldsymbol{M}\ (\mathbf{x} - \mathbf{y}))^{-0.5}$$

  - where M $\in$ $R^{d \times d}$ is a symmetric PSD matrix.

- Relation to the Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = ||\boldsymbol{Lx} - \boldsymbol{Ly}||_2 = ||\boldsymbol{L}(\boldsymbol{x} - \boldsymbol{y})||_2$$
$$= ((\mathbf{x} - \mathbf{y})^T \boldsymbol{L}^T \boldsymbol{L}\ (\mathbf{x} - \mathbf{y}))^{-0.5}$$

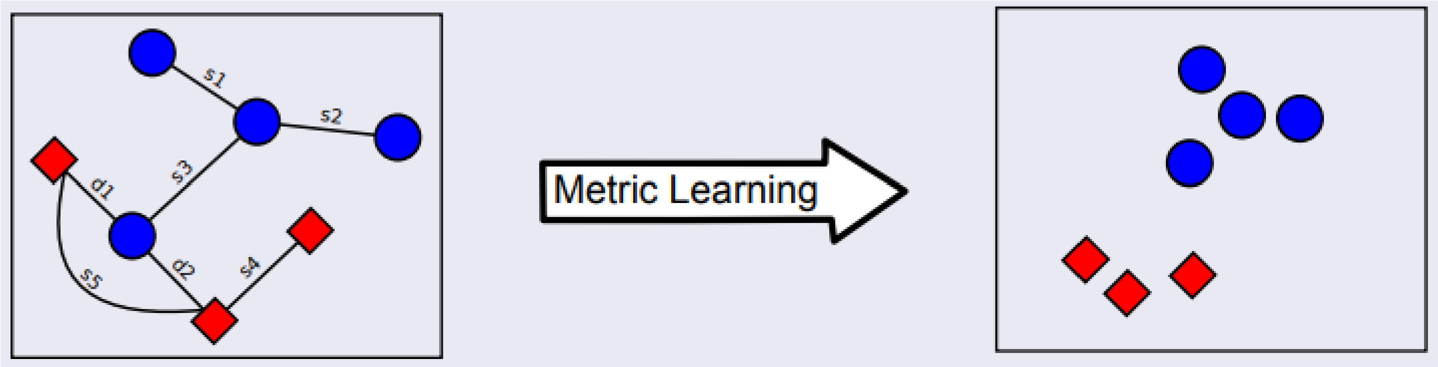
Similarity

- A (dis)similarity function is a pairwise function
  - $s(\mathbf{x}, \mathbf{y}) = 1$ (or maximum similarity) only if $\mathbf{x}=\mathbf{y}$.
  - $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$. (Symmetry)

  where $s(\mathbf{x}, \mathbf{y})$ is the similarity between points (data objects), $\mathbf{x}$ and $\mathbf{y}$.
- Bilinear similarity
  - $s(\mathbf{x}, \mathbf{y})= \mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{y}$
- Cosine similarity
  - $s(\mathbf{x}, \mathbf{y})= \mathbf{x}^{\mathrm{T}}\mathbf{y}/(||\mathbf{x}||\cdot||\mathbf{y}||)$

Metric Learning in a Nutshell

- Learning from side information
  - Must-link / cannot-link constraints:
    - S = {(x_i , x_j) : x_i and x_j should be similar},
    - D = {(x_i , x_j) : x_i and x_j should be dissimilar}.
  - Relative constraints:
    - R = {(x_i , x_j , x_k) : x_i should be more similar to x_j than to x_k}.
- Geometric intuition: learn a projection of the data



Metric Learning in a Nutshell

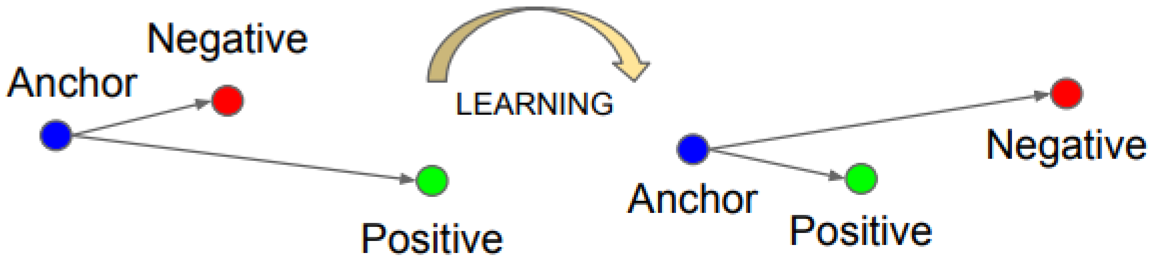
- General formulation
  - Given a metric, find its parameters $\mathbf{M}^*$ as
  - $\mathbf{M}^* = \arg\min_{\mathbf{M}} [L(\mathbf{M}, S, D, R) + \lambda R(\mathbf{M})]$,
  - where $L(\mathbf{M}, S, D, R)$ is a loss function that penalizes violated constraints, $R(\mathbf{M})$ is some regularizer on $\mathbf{M}$ and $\lambda \geqslant 0$ is the regularization parameter
- We usually have pairwise terms that represent the relations among samples in the distance metric learning loss function, also known as contrastive loss

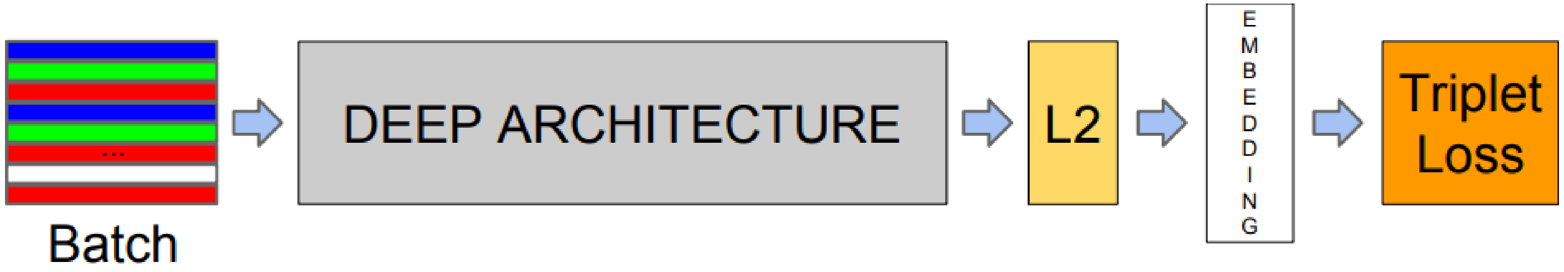
Contrastive Losses

- Triplet loss

Margin

$$\sum_{i}^{N}\left[\|f(x_i^a)-f(x_i^p)\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\alpha\right]_+$$



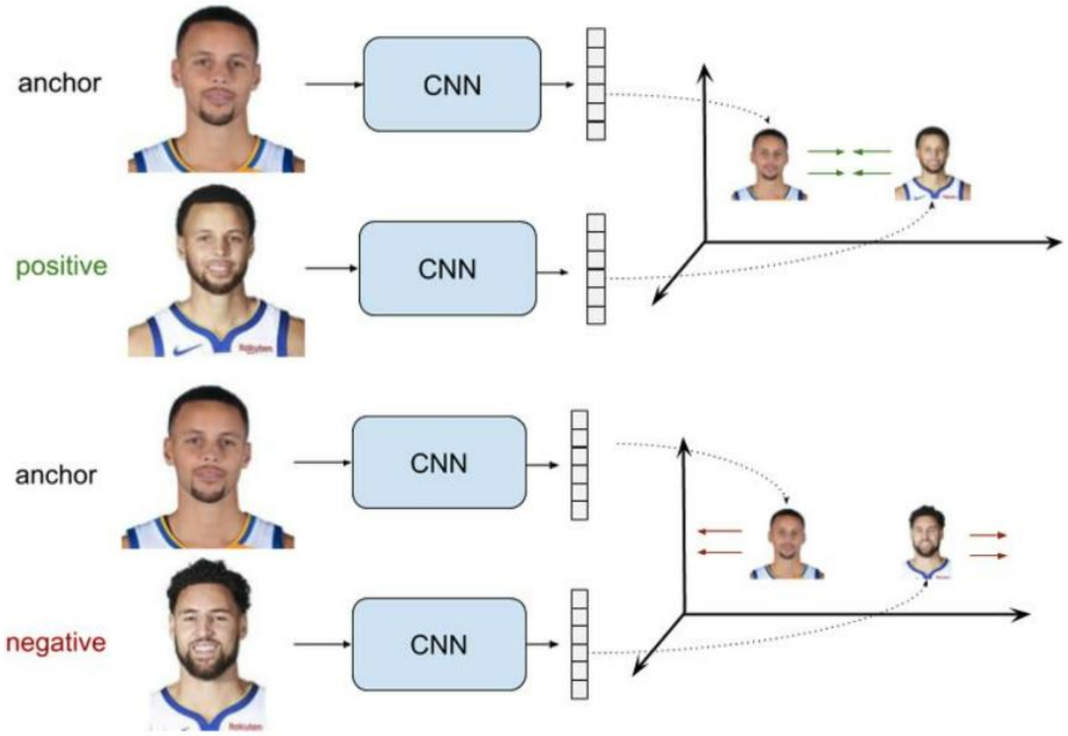


Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 815-823.

Contrastive Losses

- Ranking loss

$$L(r_0, r_1, y) = y||r_0 - r_1|| + (1 - y)\max(0, m - ||r_0 - r_1||)$$



r: representation
y=1: positive pair
y=0: negative pair
m: margin

Contrastive Losses

- Margin ranking loss
  - Usually for regression task
  - If y=1 then it assumed the first input should be ranked higher (have a larger value) than the second input, and vice-versa for y = -1.

$$l(x_1, x_2, y) = \max(0, -y * (x_1 - x_2) + m)$$

  - Example: image quality assessment, sentiment analysis

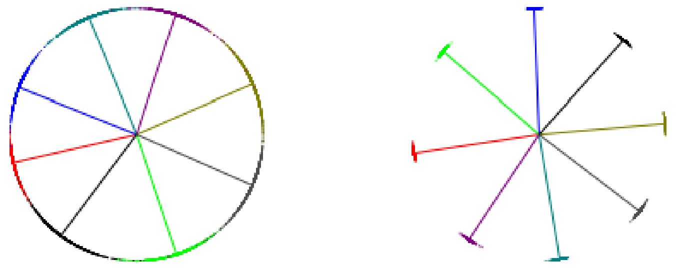
Contrastive Losses

- From softmax cross entropy loss to additive angular margin loss (arcface)
- Softmax CE

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i+b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i+b_j}}$$

- Additive angular margin loss

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}$$



(a) Softmax (b) ArcFace

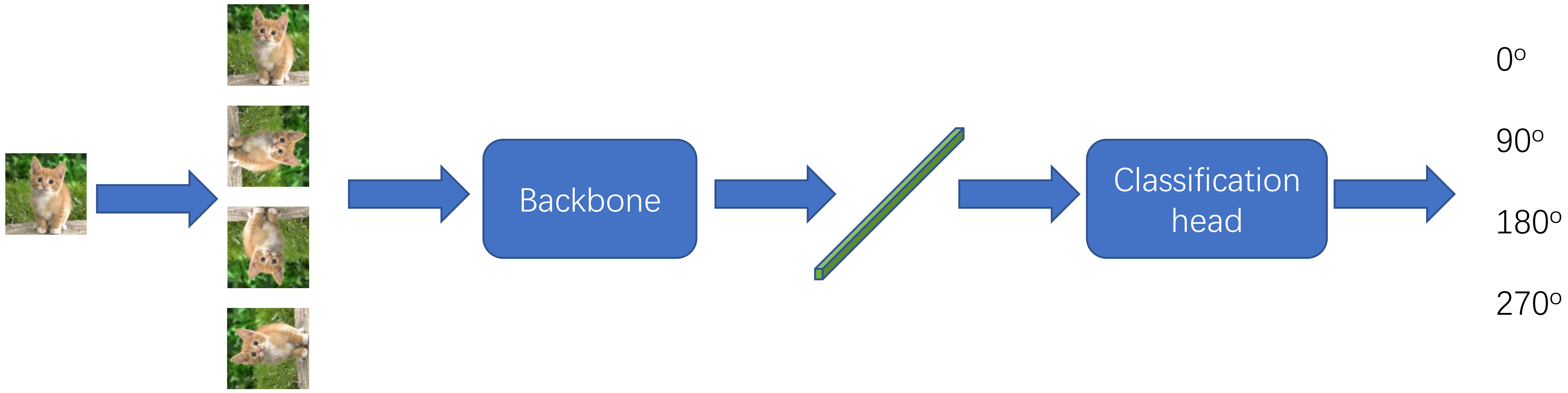
Deng J, Guo J, Xue N, et al. Arcface: Additive angular margin loss for deep face recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4690-4699.

Self-Supervised Learning

- Learn representation with self generated labels
  - Neutral representations
  - Usually for pre-training on unlabeled data
- Framework for transfer learning with self-supervised pretraining
  - Step 1: generate labels on augmented data
  - Step 2: learn backbone/encoder with supervised loss
  - Step 3: add head network for down stream tasks (usually have annotations)
  - Step 4: fine-tune the model

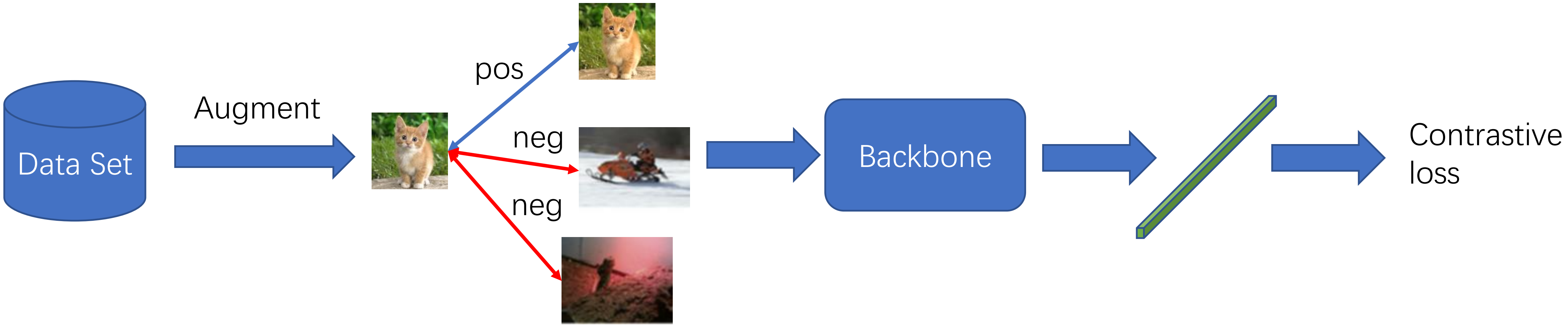
Self-Supervised Learning

- How to generate labels
  - Rotation



Self-Supervised Learning

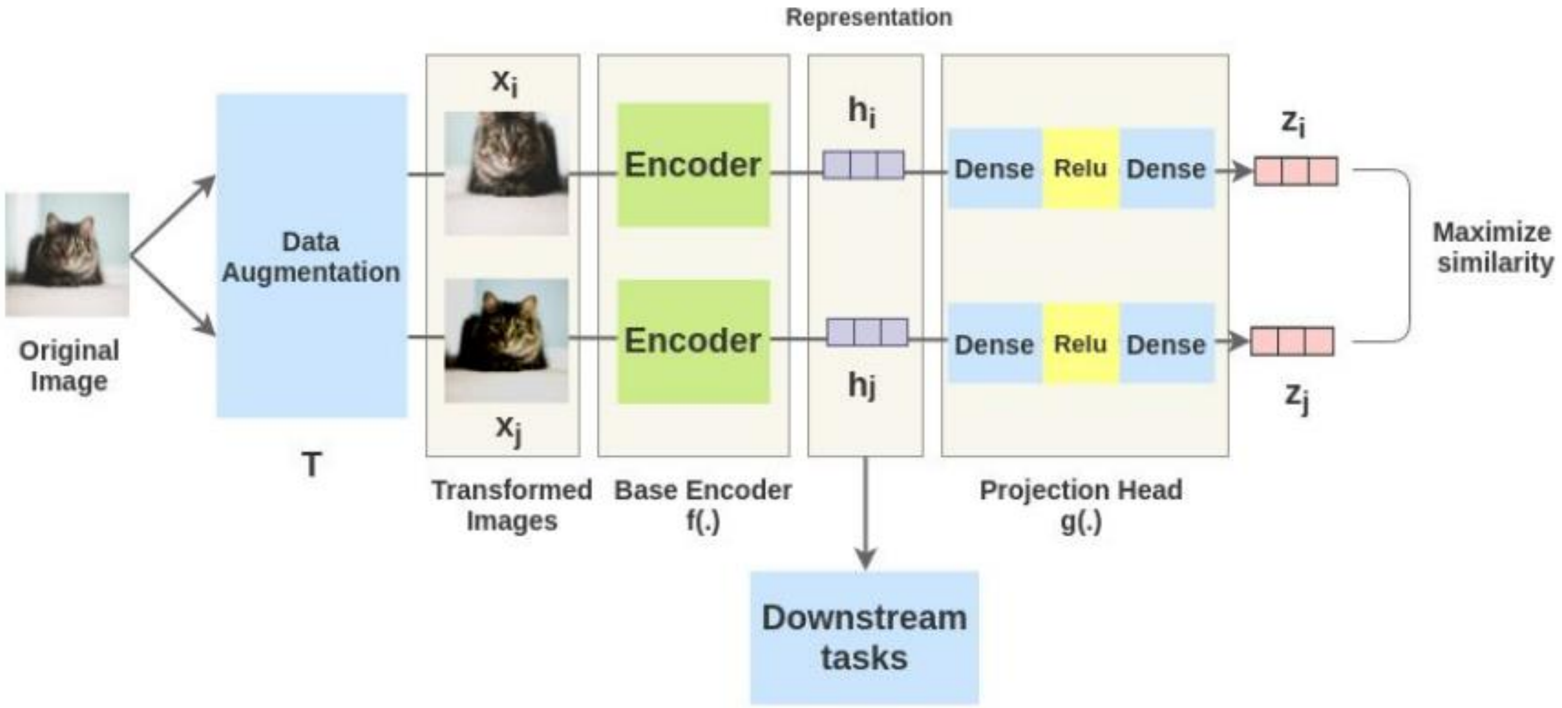
- How to generate labels
  - Generate positive and negative pairs on augmented data



Self-Supervised Learning

- Some self-supervised learning frameworks
  - A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - Momentum Contrast for Unsupervised Visual Representation Learning (MoCo)
  - Bootstrap your own latent: A new approach to self-supervised Learning (BYOL)

SimCLR

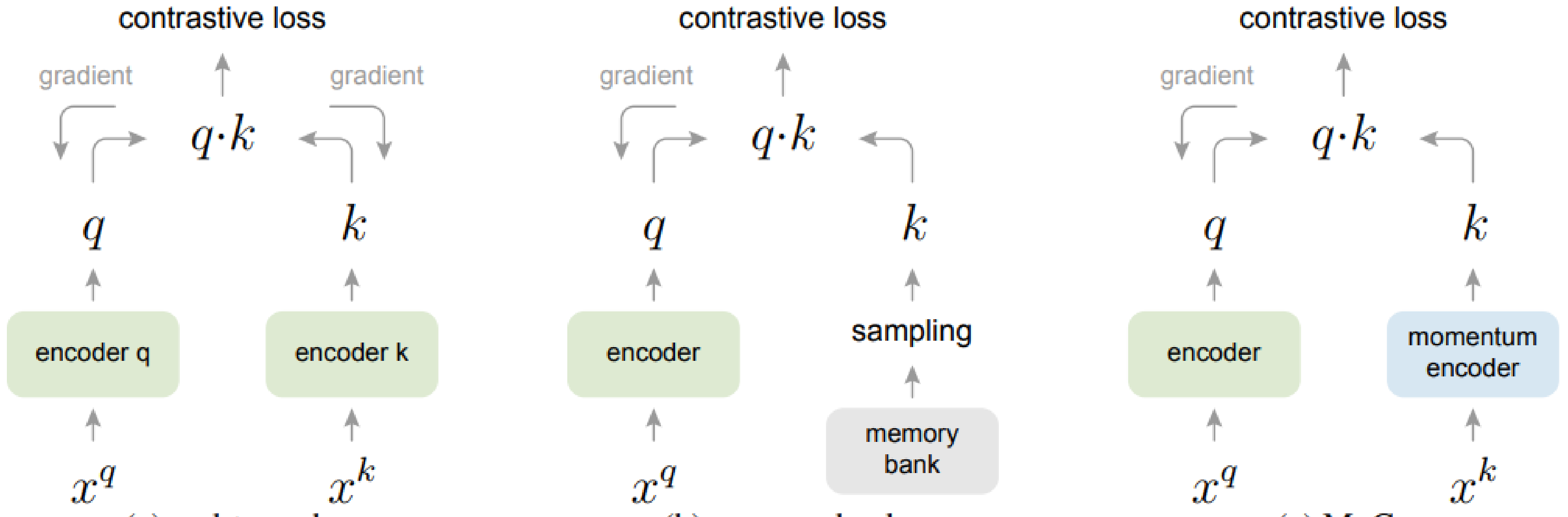


$$-\frac{1}{|\mathcal{X}_+|}\log\frac{\sum_{\mathbf{x}'\in\mathcal{X}_+}\exp\left(\mathrm{sim}(f_{\Theta}(\mathbf{x}),f_{\Theta}(\mathbf{x}'))/\tau\right)}{\sum_{\mathbf{x}'\in(\mathcal{X}_+\cup\mathcal{X}_-)}\exp\left(\mathrm{sim}(f_{\Theta}(\mathbf{x}),f_{\Theta}(\mathbf{x}'))/\tau\right)}$$

Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//International conference on machine learning. PMLR, 2020: 1597-1607.

MoCo

- Enlarge the set of negative samples
- Momentum encoder
  - $\boldsymbol{\theta}_k \leftarrow m\boldsymbol{\theta}_k + (1 - m)\boldsymbol{\theta}_q$



(a) end-to-end (b) memory bank (c) MoCo

He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 9729-9738.

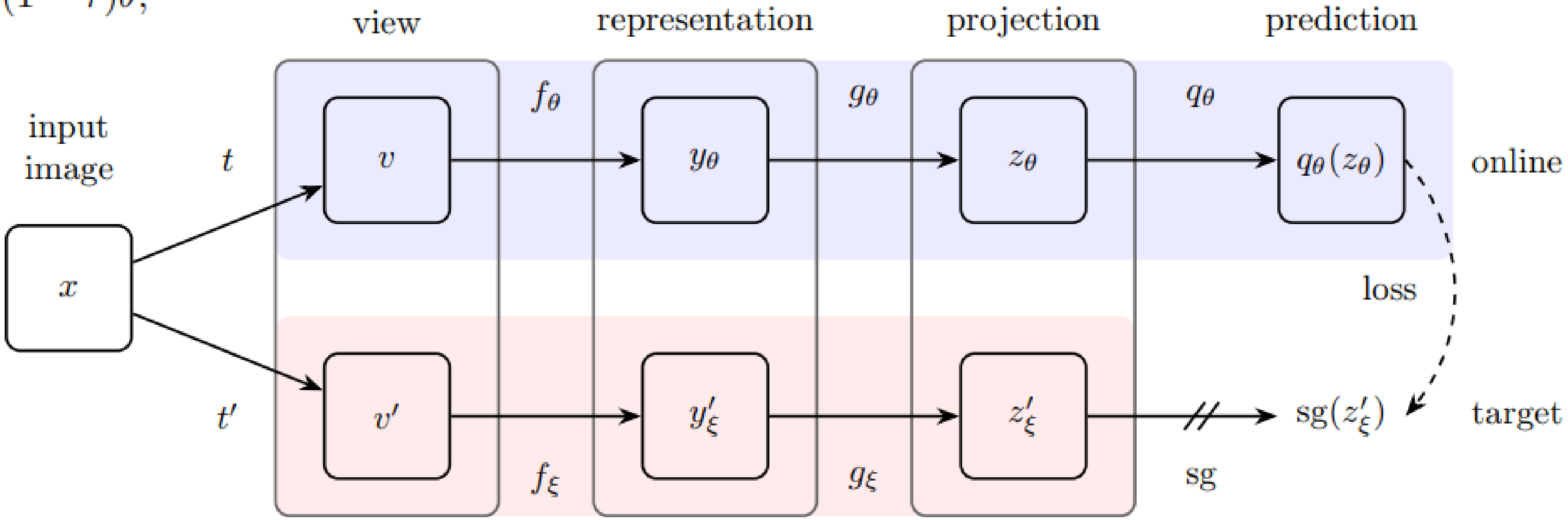
BYOL

- No negative samples

$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\|q_\theta(z_\theta)\right\|_2 \cdot \left\|z'_\xi\right\|_2}$$

$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\texttt{BYOL}}_{\theta,\xi}, \eta\right),$$

$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$



Grill J B, Strub F, Altché F, et al. Bootstrap your own latent: A new approach to self-supervised learning[J]. arXiv preprint arXiv:2006.07733, 2020.