数据挖掘与应用

深度聚类

授课教师：周晟

浙江大学　软件学院

2021.10

教学内容

1. 深度聚类
2. 初探深度聚类
3. 嵌入式聚类系列
4. 对比聚类系列
5. 伪标签系列
6. 基于聚类的表征学习
7. 其他类型数据的聚类

深度聚类

研究动机

经典聚类方法聚焦于给定数据特征在特征空间进行聚类，而真实场景中很多数据难以使用简单的特征进行描述。对这些数据进行聚类依赖于好的特征表示。

对比 K-means 和 Spectral Clustering，说明了将数据进行有效降维对于聚类的意义。然而 Spectral 仅能处理简单的，可计算相似度的数据，难以应用于复杂数据。

深度学习

结合深度学习进行聚类的优势：

1. 有效提取复杂类型数据的特征
2. 可用于大规模数据集
3. 可建模非线性依赖关系

深度表征学习

定义

表征学习是指将数据表示成低维向量的过程。

常见的表征学习方法包括：

1. PCA,LDA
2. Lapalacian
3. Manifold Learning
4. Auto-encoder
5. Deep Generative Model
6. Self-supervised Learning
7. Contrastive Learning

常见的深度聚类应用

1. 图像聚类
   1. 智能相册
   2. 医学诊断
   3. 智慧交通
2. 文档聚类
   1. 图书分类
   2. 文档摘要生成
3. 音视频聚类
   1. 音乐分类
   2. 音乐推荐
4. 网络聚类
   1. 社交推荐
   2. 社区发现

深度聚类的主要组件

表征学习

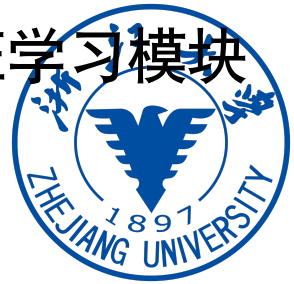
将复杂的数据类型转化为传统聚类算法更容易接受的特征向量形式。

相比于传统的表征学习，深度聚类中的表征学习模块更强调对复杂类型数据的建模能力。

聚类模块

对表征学习模块得到的样本向量进行聚类，也可用于指导表征学习。

相比于传统的聚类方法，深度聚类中的聚类模块更强调对表征学习模块的影响。



深度聚类的主要组件

表征学习模块包含了模型的骨干网络（Backbone），它能够将输入的原始图像数据转换为更适合于聚类的表征数据，不仅是减少数据量，更是希望使得相同类的样本在特征空间中接近，而不同类的样本分离。

骨干网络	方法
自动编码器（AE）	DEC
卷积神经网络（CNN）	DAC
生成对抗网络（GAN）	GAN
变分自编码器（VAE）	VAE
残差网络（ResNet）	CC、IDFD、SCAN、RUC、TSUC、IIC、DeepCluster

图：骨干网络总结

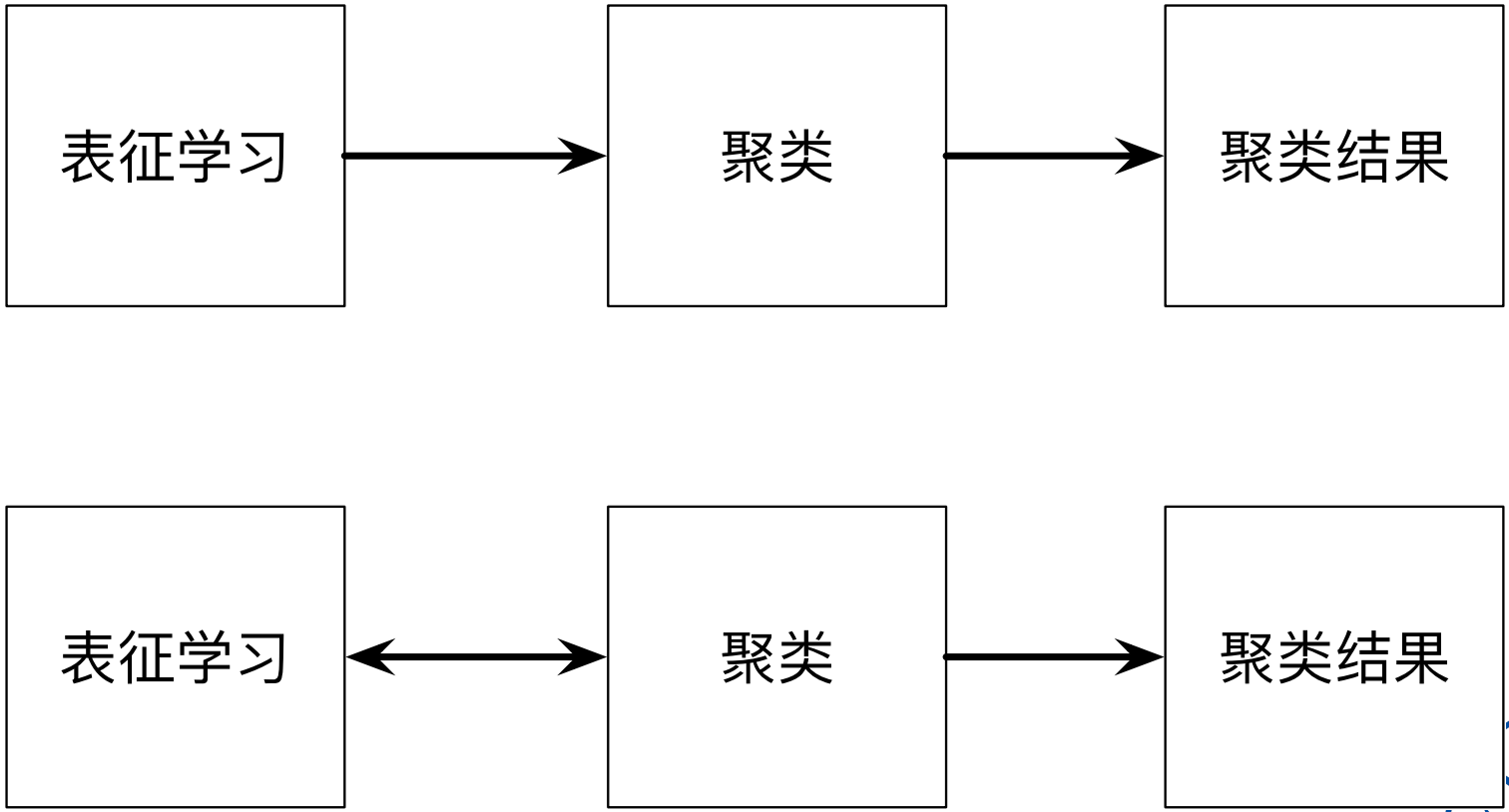
深度聚类的主要组件

聚类模块是深度聚类算法的核心。通过表征学习模块，我们将原始数据转换成了低维的表征向量，而如何利用表征将对应的样本进行聚类，这就是聚类模块要考虑的问题。

聚类模块	方法
K-Means	IDFD
Multiple Experts	MiCE
Construct Graph	Kingdra
Student's t-Distribution	DEC、AGAE
Network Layers	ADC、CC、DAC、SCAN

图：聚类模块总结

深度聚类框架



深度聚类主要挑战

平凡解（trival solution），即由于设计不当导致 Loss 函数存在全局最优解，比如将所有样本聚类到一个簇中。这样做可以使 Loss 最小，但显然是无意义的。

一种可行的解决方案是在 Loss 函数中添加一项约束：

1. 每个类的样本数量均匀
2. 避免出现所有样本归为一个类
3. 其他数据相关的约束

深度图像聚类数据集

MNIST 数据集 (Mixed National Institute of Standards and Technology database) 是一个小型手写数字数据集，包含 60000 个样本的训练集和 10000 个样本的测试集。每个样本是 $1 \times 28 \times 28$ 的单通道灰度图片数据。

图: MNIST 数据集

深度图像聚类数据集

CIFAR-10 是一个用于识别普适物体的小型数据集。一共包含 10 个类别的 RGB 彩色图片：

飞机（airplane）、汽车（automobile）、鸟类（bird）、猫（cat）、鹿（deer）、狗（dog）、蛙类（frog）、马（horse）、船（ship）和卡车（truck）。

每个样本是 $3 \times 32 \times 32$ 的三通道 RGB 图片。数据集中一共有 50000 张训练图片和 10000 张测试图片。

深度图像聚类数据集

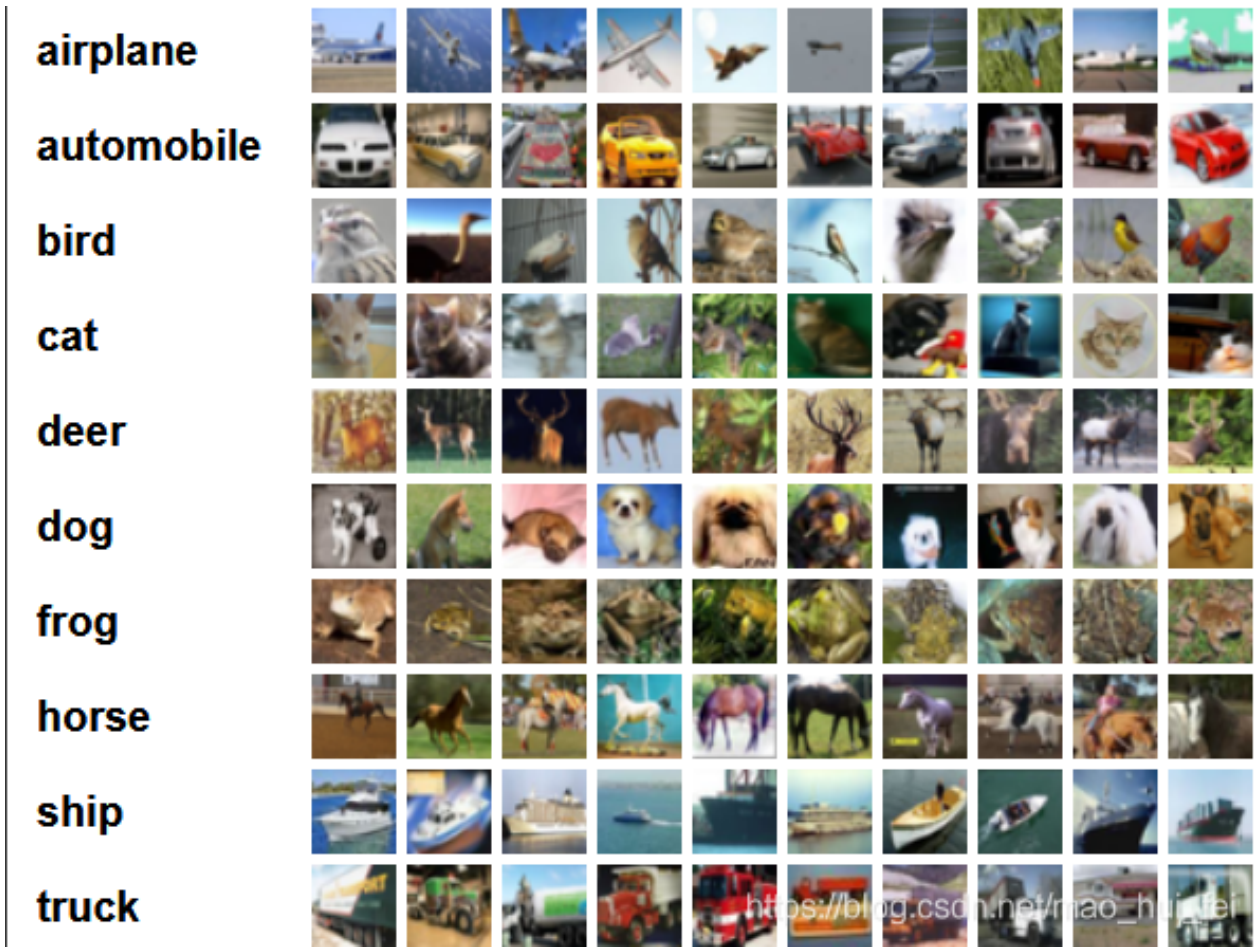


图: CIFAR-10 数据集

深度图像聚类数据集

CIFAR-100 与 CIFAR-10 有着同样的图片，只是它分得更为细致。

CIFAR-100 有 100 个类，每个类包含 600 张图片，其中 500 张为训练图片，100 张为测试图片。而这 100 个类又可以被分为 20 个超类。

每个图像都带有一个“精细”标签（它所属的类）和一个“粗糙”标签（它所属的超类）。

深度图像聚类数据集

ImageNet 数据集是按照 WordNet 架构组织的大型图像数据集，包含约 1500 万张图片和 2.2 万个类。图片的分辨率不是固定的，存在各种清晰度，需要自己进行处理。当然，通道数还是 RGB 三通道。

实际使用中，由于 ImageNet 过于庞大，所以一般用的都是 ImageNet 的子集，如 ImageNet-Dogs 和 Tiny ImageNet 等。

深度图像聚类数据集

ImageNet-Dogs 是 ImageNet 的一个子集，在一次 Kaggle 比赛中作为数据集。训练集包含 10222 张图像，测试集包含 10357 张图像，都为 RGB 三通道，但是具有不同的高度和宽度。

数据集中有 120 种狗，包括拉布拉多犬、贵宾犬、腊肠犬、萨摩耶犬、哈士奇犬、吉娃娃犬和约克郡梗等等。

图: ImageNet-Dogs

深度图像聚类数据集

Tiny ImageNet 是一个微型的 ImageNet，来自斯坦福大学 cs231N 的课程项目。

Tiny ImageNet 共 200 个类，每个类有 500 个训练样本，50 个验证样本，50 个测试样本，每个样本大小是 $3 \times 64 \times 64$。

深度图像聚类的评价

目前，在深度图像聚类领域较常用的评价指标有：

- 聚类精度 ACC

$$ACC = \max_{m} \frac{\sum_{i=1}^{N} \mathbf{1}\{y_i = m(c_i)\}}{N}$$

- 标准化互信息 NMI

$$NMI(Y, C) = \frac{I(Y, C)}{\frac{1}{2}[H(Y) + H(C)]}$$

- 调整兰德系数 ARI

$$ARI = \frac{RI - E[RI]}{\max RI - E[RI]}$$

Deep Cluster

Deep Clustering for Unsupervised Learning of Visual Features[CBJD18] (ECCV 2018) 是经典的深度聚类方法之一。

基本思想

好的表征学习应该应当保留样本的局部特征，也需要考虑全局的语义特征，例如聚类等信息。而好的表征也能帮助进行更好的聚类。

研究目标

对于数据集 $X = \{x_1, x_2, ..., x_N\}$，若每个样本 x_i 都有相对应的标签 y_i。如何训练特征提取器 f_θ 和分类器 g_W，使其具有较好的特征提取和分类的能力？

Deep Cluster

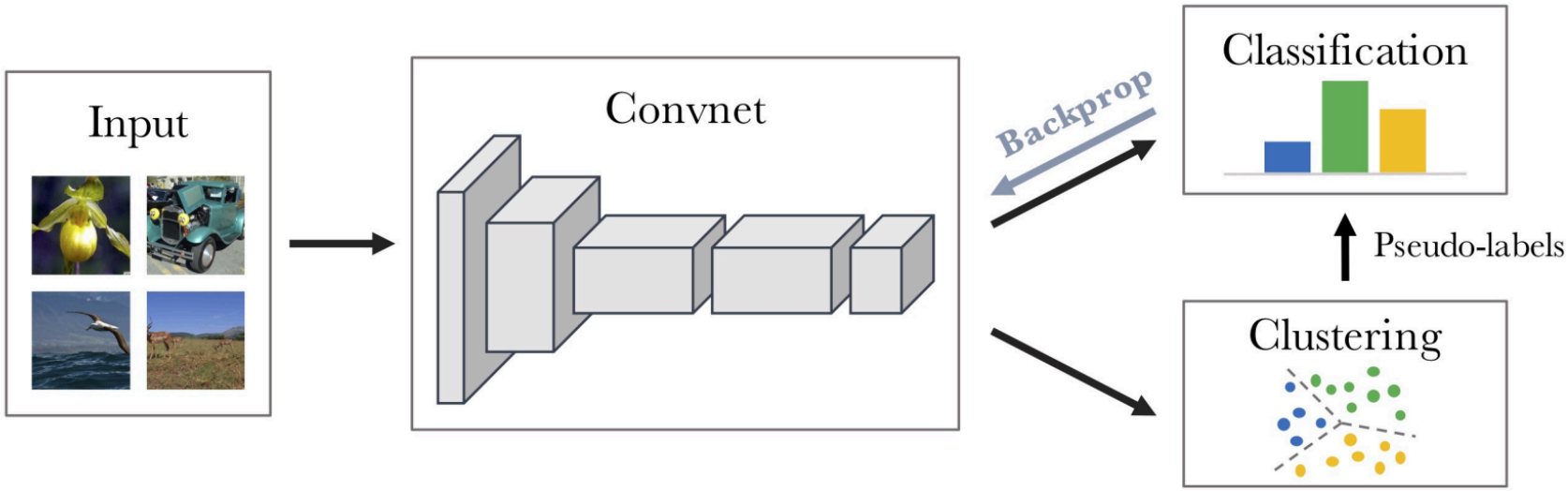


图: DeepCluster 模型架构

Deep Cluster

优化目标

$$\min_{\theta,W} \frac{1}{N} \sum_{i=1}^{N} \ell(g_W(f_\theta(x_i)), y_i)$$

其中 ℓ 是某种 Loss 形式，比如最常用的 MSE Loss。

然而在无监督场景下难以获得真实标签。

Deep Cluster

那么如何获得标签呢？

在已经有特征提取器 f_θ 的前提下，Deep Cluster 选择了一种最简单的方法：K-means。

形式上来说，K-means 通过优化下式来学习 $k \times d$ 的簇中心矩阵和每个图像 i 的聚类分配 y_i：

$$\min_{C \in \mathcal{R}^{k \times d}} \frac{1}{N} \sum_{i=1}^{N} \min_{y_i \in \{0,1\}^k} \|f_\theta(x_i) - C_{y_i}\|_2^2 \quad s.t. \quad y_i^T 1_k = 1$$

Deep Cluster

由此，Deep Cluster 在以下两个步骤之间迭代：

- 通过 K-means 获得当前样本特征所对应的 pseudo label
- 通过 pseudo label 优化目标从而更新模型的参数

Deep Cluster

Method	Classification		Detection		Segmentation	
	FC6-8	ALL	FC6-8	ALL	FC6-8	ALL
ImageNet labels	78.9	79.9	–	56.8	–	48.0
Random-rgb	33.2	57.0	22.2	44.5	15.2	30.1
Random-sobel	29.0	61.9	18.9	47.9	13.0	32.0
Pathak *et al.* [38]	34.6	56.5	–	44.5	–	29.7
Donahue *et al.* [20]*	52.3	60.1	–	46.9	–	35.2
Pathak *et al.* [27]	–	61.0	–	52.2	–	–
Owens *et al.* [44]*	52.3	61.3	–	–	–	–
Wang and Gupta [29]*	55.6	63.1	32.8†	47.2	26.0†	35.4†
Doersch *et al.* [25]*	55.1	65.3	–	51.1	–	–
Bojanowski and Joulin [19]*	56.7	65.3	33.7†	49.4	26.7†	37.1†
Zhang *et al.* [28]*	61.5	65.9	43.4†	46.9	35.8†	35.6
Zhang *et al.* [43]*	63.0	67.1	–	46.7	–	36.0
Noroozi and Favaro [26]	–	67.6	–	53.2	–	37.6
Noroozi *et al.* [45]	–	67.7	–	51.4	–	36.6
DeepCluster	**70.4**	**73.7**	**51.4**	**55.4**	**43.2**	**45.1**

图: Deep Cluster 表征学习结果

Deep Cluster

Deep Cluster 如何避免 trivial solution ?

DEC

Deep Embedding Clustering(DEC) 是最具代表性的深度聚类算法之一。它将样本表征的学习和聚类分配的学习统一到了一个框架中，共同学习。

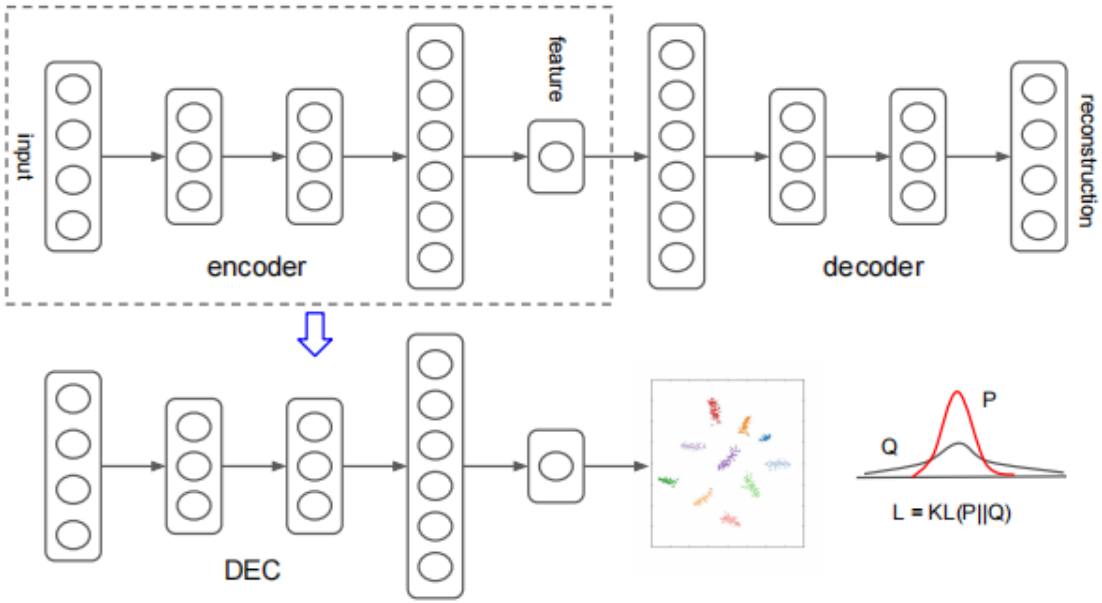


图: DEC 框架

DEC——网络架构

自动编码器

自动编码器（AutoEncoder，AE）。它由一个输入层、一个隐藏层和一个输出层组成。输入 x 在经过编码和解码后得到输出 z，我们希望 z 尽可能与 x 相同，隐藏层的输出 y 可以看做 x 的特征向量。

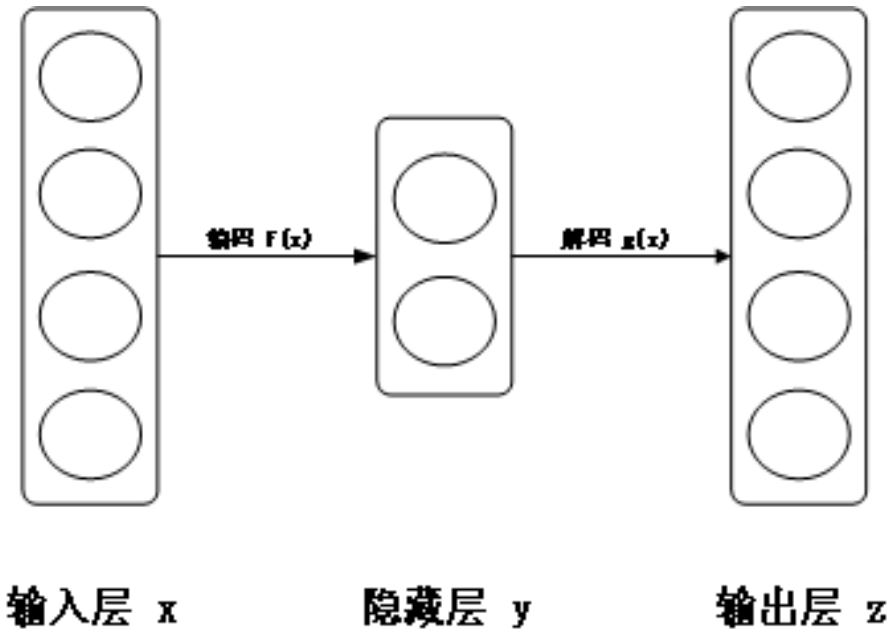


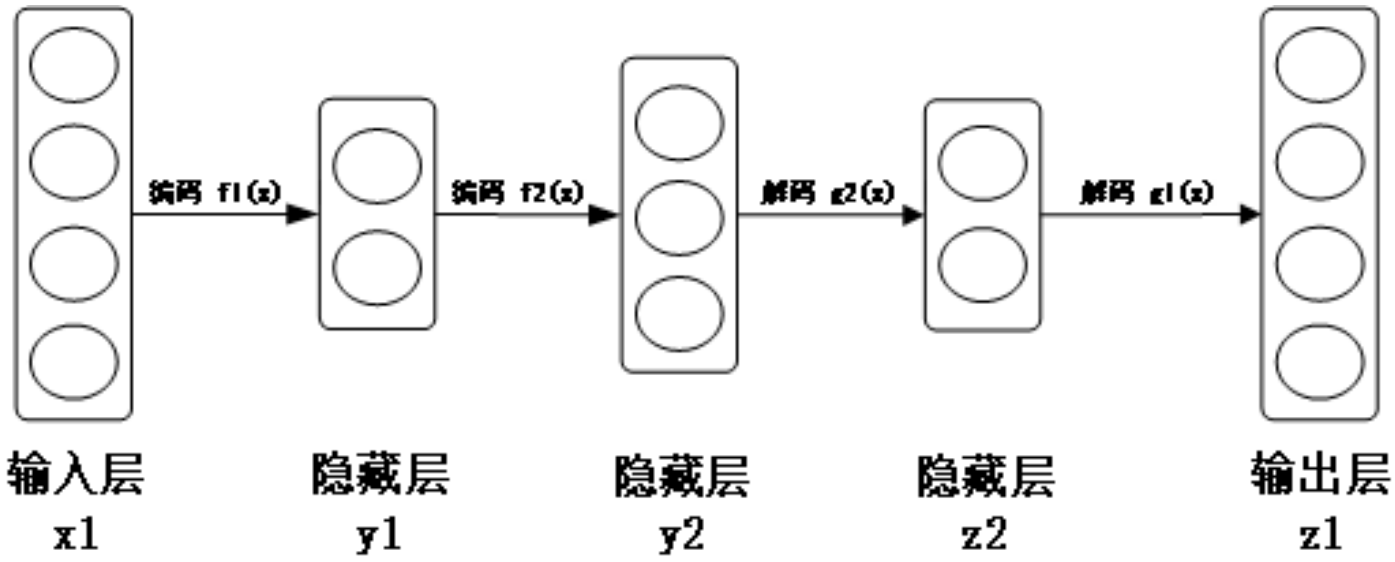
图: AE 网络结构

DEC——网络架构

堆叠自编码器

堆叠自动编码器（Stacked AutoEncoder，SAE）由多个 AE 堆叠而成。

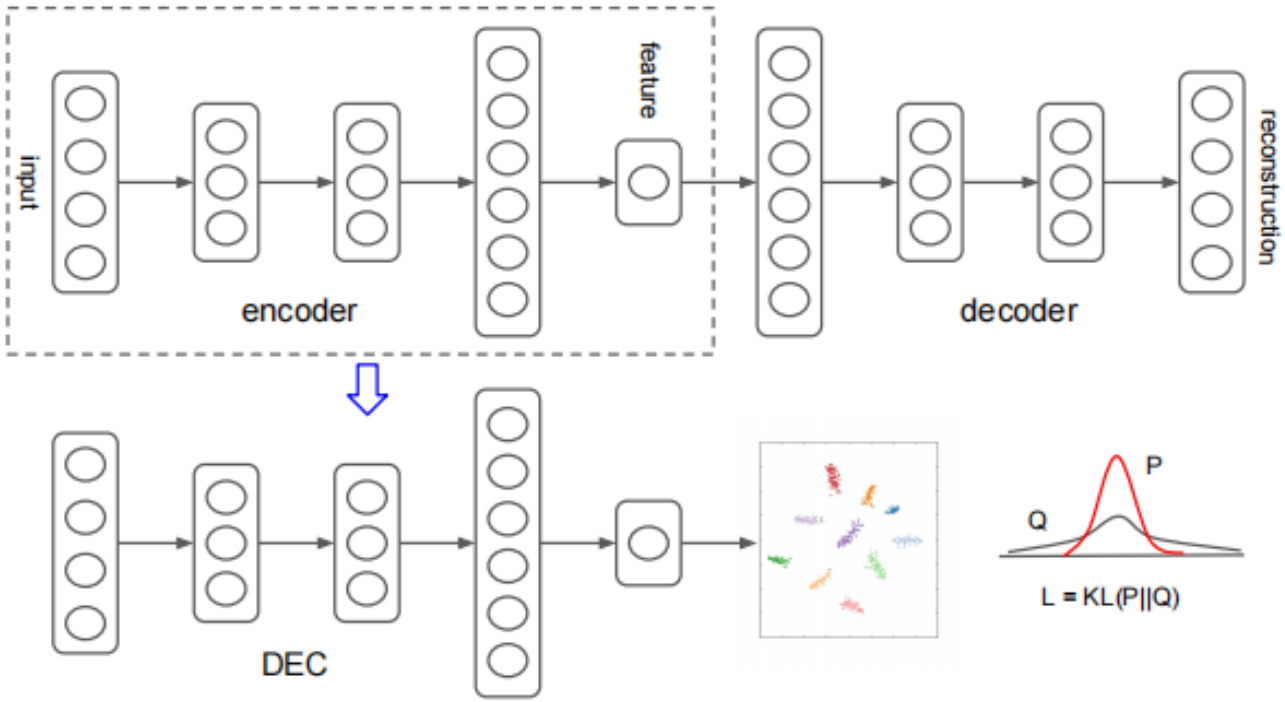
构造时先训练第一个 AE，将隐藏层输出 y_1 作为第二个 AE 的输入 x_2，训练第二个 AE，直到所有 AE 都训练完成，最终对堆叠而成的整个网络进行微调。



图：两个 AE 堆叠而成的 SAE

DEC——网络架构

DEC 的网络架构是基于 SAE 构造的：它将 SAE 的编码器部分拿了出来，作为样本的特征提取器，也就是 DEC 的主要网络结构。



图：DEC 网络架构

DEC——主要思想

1. 初始化网络参数 θ
2. 输入样本 x，提取其特征 $z = f_\theta(x)$
3. 根据特征 z 和聚类中心 μ 计算样本的软分配 q
4. 根据软分配 q 构造目标分配 p
5. 根据软分配和目标分配计算 loss，回传更新网络参数和聚类中心
6. 重复步骤 $2 \sim 5$，直到满足退出条件

DEC——参数初始化

- 网络参数的初始化：
  预训练 SAE，最小化重构 loss，将此时 SAE 的 encoder 部分提取出，作为 DEC 的网络结构。

- 聚类中心的初始化：
  对样本特征 z 使用 K-means 算法，将 K-means 算法产生的聚类中心作为 DEC 初始化的聚类中心 μ。

DEC——Student's t 分布

Student's t 分布：如果我们有两个独立的随机变量 U 和 V，U 服从标准正态分布 $\mathcal{N}(0,1)$，V 服从自由度为 α 的卡方分布 $\mathcal{X}_{\alpha}^{2}$。那么 $\frac{U}{\sqrt{V/\alpha}}$ 服从的分布是一个自由度为 α 的 t 分布。

t 分布概率密度函数：

$$f_t(x) = \frac{\Gamma(\frac{\alpha+1}{2})}{\sqrt{\alpha\pi}\Gamma(\frac{\alpha}{2})}\left(1+\frac{x^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}$$

其中 Γ 为伽马函数。

DEC——目标函数

- 根据特征 z 和聚类中心 μ 计算样本的软分配 q:

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + \|z_i - \mu_{j'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}$$

其中 α 为 Student's t 分布的自由度，DEC 中取 $\alpha = 1$。

DEC 假设特征与聚类中心之间的距离 $\|z - \mu\|^2$ 满足 Student's t 分布，即把 $\|z - \mu\|^2$ 当做 Student's t 分布中的 x。

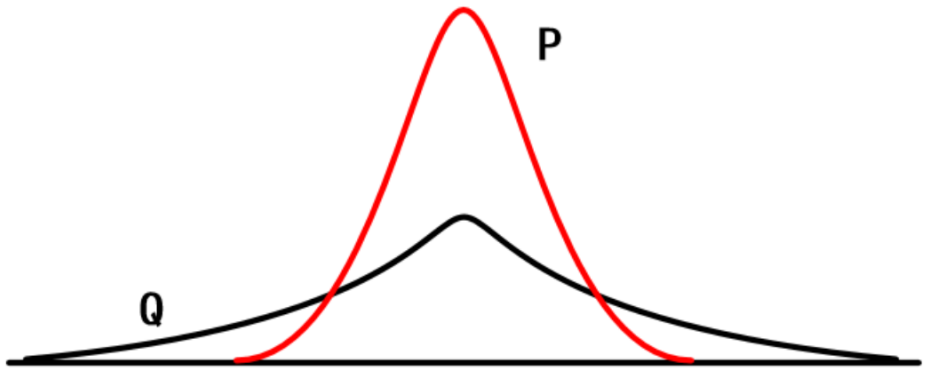
而前面的系数 $\frac{\Gamma(\frac{\alpha+1}{2})}{\sqrt{\alpha\pi}\Gamma(\frac{\alpha}{2})}$ 是一个关于 α 的值，可以在分子和分母中上下约掉。

DEC——目标函数

- 根据软分配 q 构造目标分配 p:

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}}$$

其中 $f_j = \sum_i q_{ij}$ 是软聚类频率，有利于类别平衡，避免平凡解。



而 q^2 的作用是使软分配 q 中概率大的在目标分配 p 中越大，小的越小。

DEC——目标函数

- Loss 函数：

$$L = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

KL 散度又称相对熵，可以用来衡量两个分布之间的差异，两者差异越小，KL 散度越小。当两分布一致时，其 KL 散度为 0。

$$\begin{aligned} D_{KL}(p\|q) &= H(p,q) - H(p) \\ &= -\sum_x p(x) \log q(x) - \sum_x -p(x) \log p(x) \\ &= -\sum_x p(x) \log \frac{q(x)}{p(x)} \end{aligned}$$

选用 KL 散度作为 Loss 函数是希望软分配 q 尽可能向目标分配 p 靠拢。

DEC——实验效果

Table 2. Comparison of clustering accuracy (Eq. 10) on four datasets.

Method	MNIST	STL-HOG	REUTERS-10k	REUTERS
k-means	53.49%	28.39%	52.42%	53.29%
LDMGI	84.09%	33.08%	43.84%	N/A
SEC	80.37%	30.75%	60.08%	N/A
DEC w/o backprop	79.82%	34.06%	70.05%	69.62%
DEC (ours)	**84.30%**	**35.90%**	**72.17%**	**75.63%**

图: DEC 实验效果

DEC——方法分析

DEC 的精髓在于辅助目标分配的构造，其本质是使软分配中概率大的越大，小的越小，同时还包括了类大小的均衡约束。

当然，DEC 也存在着一些不足之处：

- 网络结构简单
- 受参数初始化影响大

IDEC

Improved Deep Embedding Clustering(IDEC) 在 DEC Loss 的基础上增加了一项重构 Loss，取得了一些略微的效果提升。

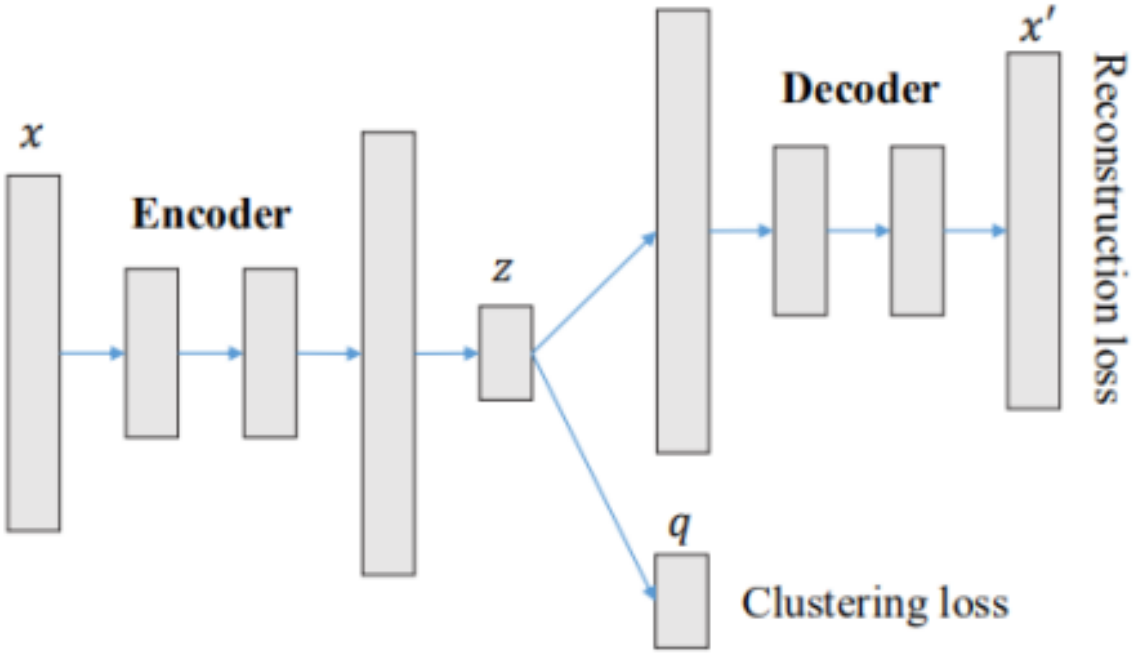


图: IDEC 网络架构

对比学习

对比学习（Contrastive Learning）是自监督表征学习中的一类，它的核心思想是在特征空间中将样本分别与正例样本和负例样本进行对比，以此来学习样本的特征表示。

简单来说，对比学习希望样本与正例样本之间的相似度最大，与负例样本之间的相似度最小。

对比学习

对任意样本数据 x，对比学习的目标是学习一个编码器 f 使得：

$$score(f(x), f(x^+)) >> score(f(x), f(x^-))$$

其中，x^+ 为与 x 相似的样本，称为正样本，两者组成正对。x^- 为与 x 不相似的样本，称为负样本，两者组成负对。

score 是一个度量函数，用来衡量样本间的相似度。

对比学习

显而易见，对比学习的重点和难点在于如何在无监督的设置下构造正负样本。

常见的策略有以下两种：

- 第一种是利用聚类结果作为伪标签来指导正负样本对的构造。
- 第二种更直接、更常用的方法是将每个样本看作一个类，通过数据增强构造样本对。具体而言，正样本对由同一样本的两个数据增强视图组成，而其余的样本对都被视为负样本对。

数据增强

以图像数据为例，数据增强（Data Augmentation）是在原始图像上进行一系列图像变换，使原始图像变为语义等价但表现形式不同的图像。

换句话说，经过数据增强后的图像在人眼中表达的是和原始图像相同的东西，但在欠训练的模型中，两者是非常不同的——这也正是我们使用数据增强的一个目的——增强鲁棒性和泛化能力，使模型的判别能力向人靠近。

数据增强

数据增强在形式上可以分为两种：

- 离线增强：直接在数据集上执行，为每一张图片构造多个数据增强版本，合并后成为一个数据量为原先若干倍的大数据集。这种方法常常用于数据集较小的情况。
- 在线增强：训练过程中对于当前 batch 的数据进行增强。对于同一个 batch，不同 epoch 中数据增强产生的图片都是不同的。在线增强往往用于一些较大的数据集——它们的数据量无法接受线性级别的增长。实际中，在线增强使用的比较频繁，因此也有许多机器学习框架都已经支持了在线增强的方式。

常用的图像变换

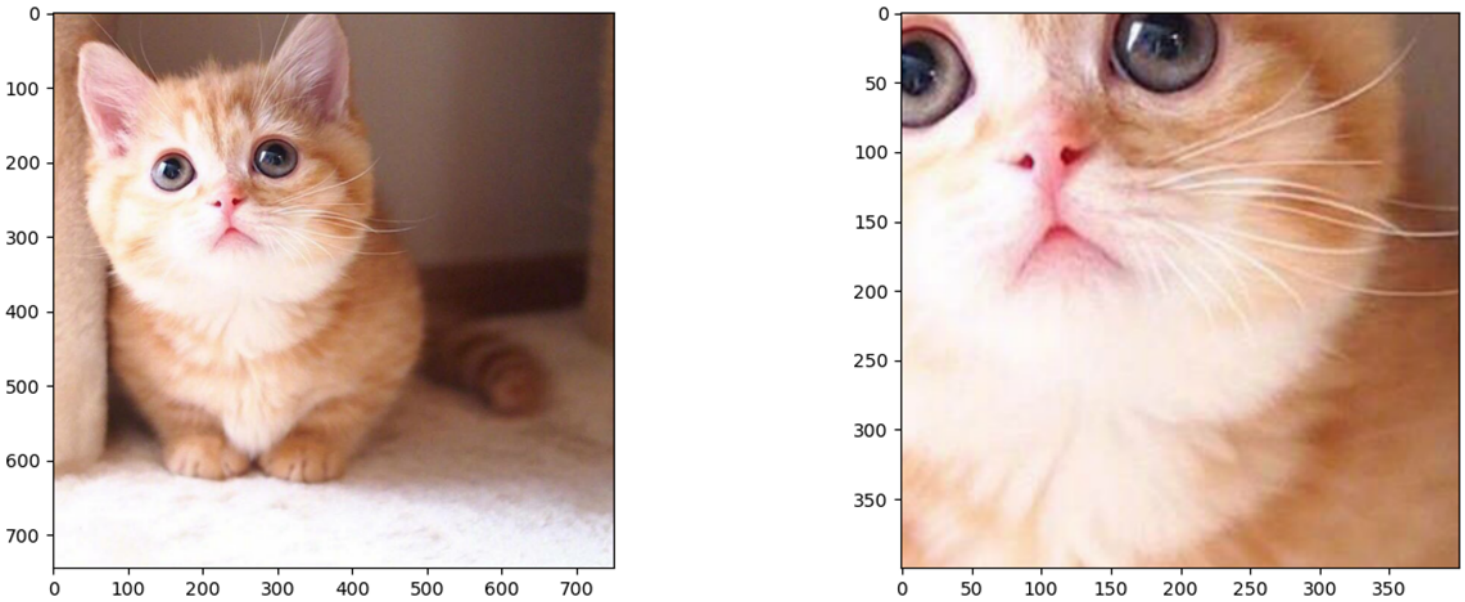
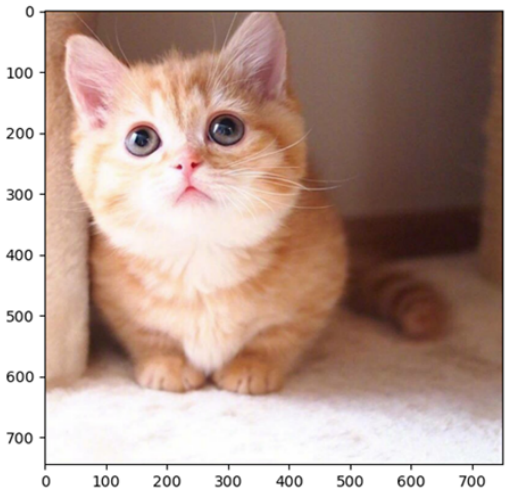
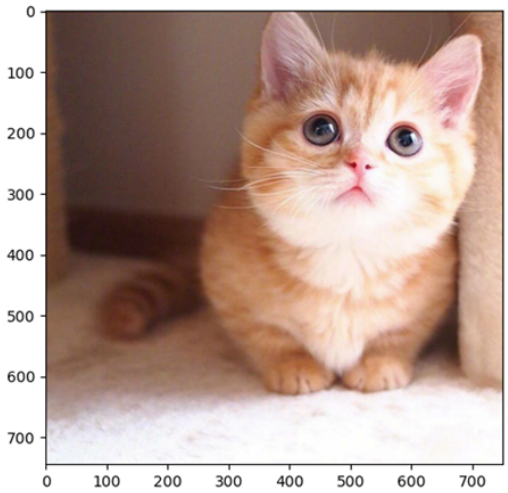


图: 随机裁剪

常用的图像变换





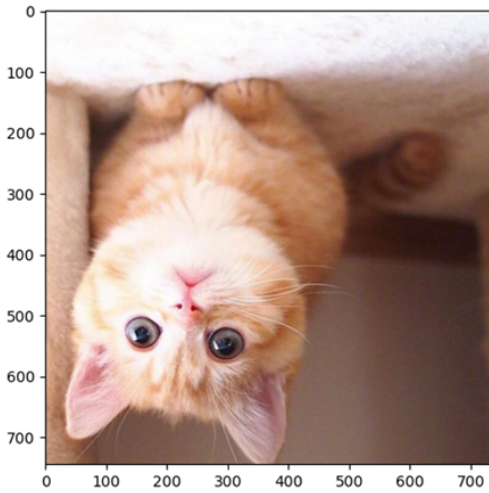


图: 水平和竖直翻转

常用的图像变换

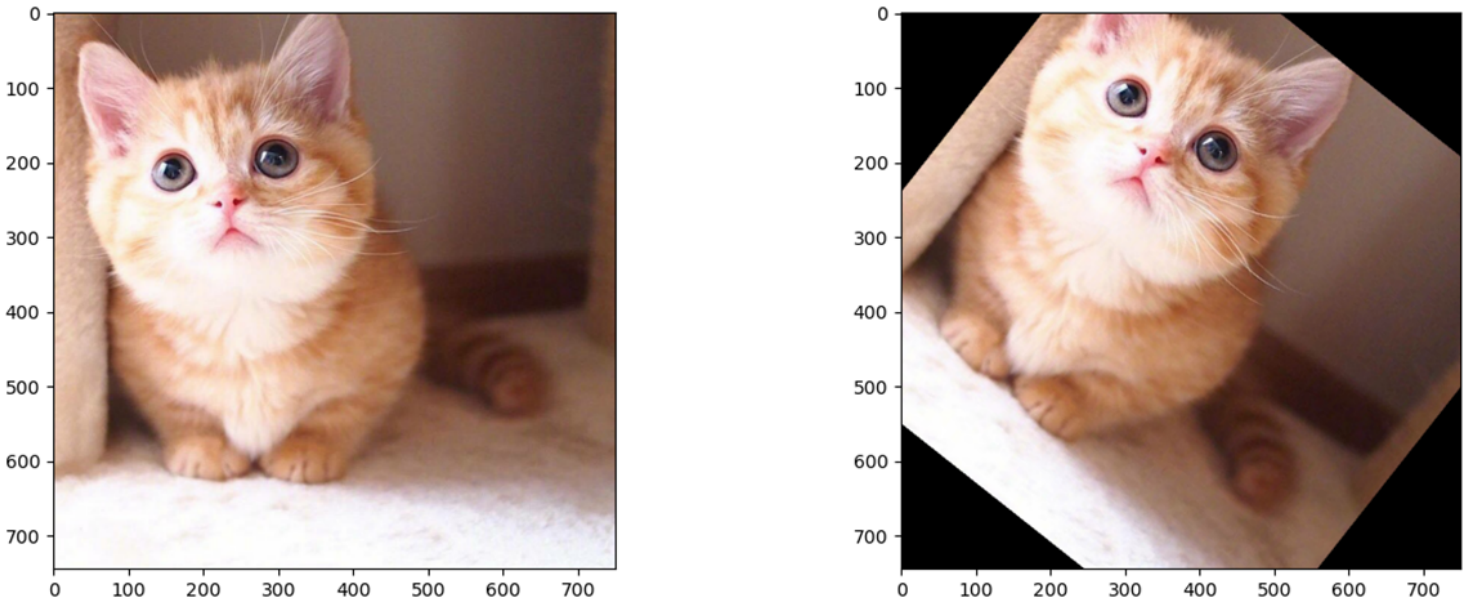


图: 随机旋转

常用的图像变换

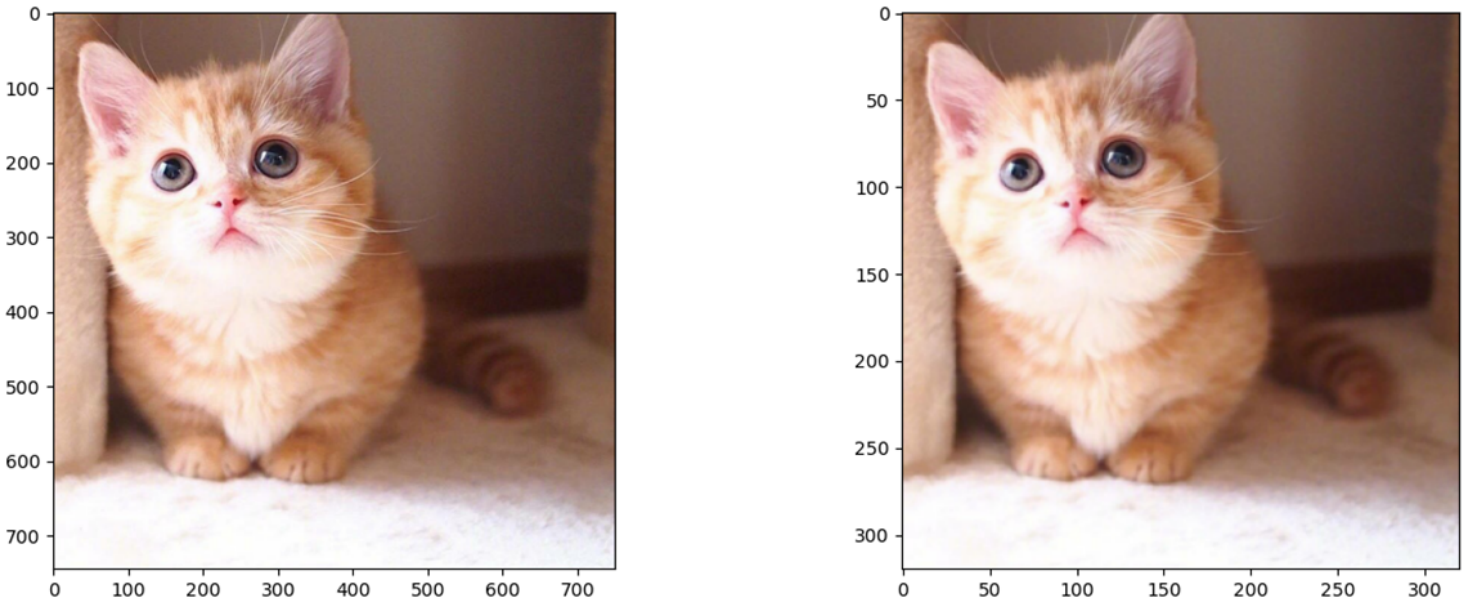


图: 图像缩放

常用的图像变换

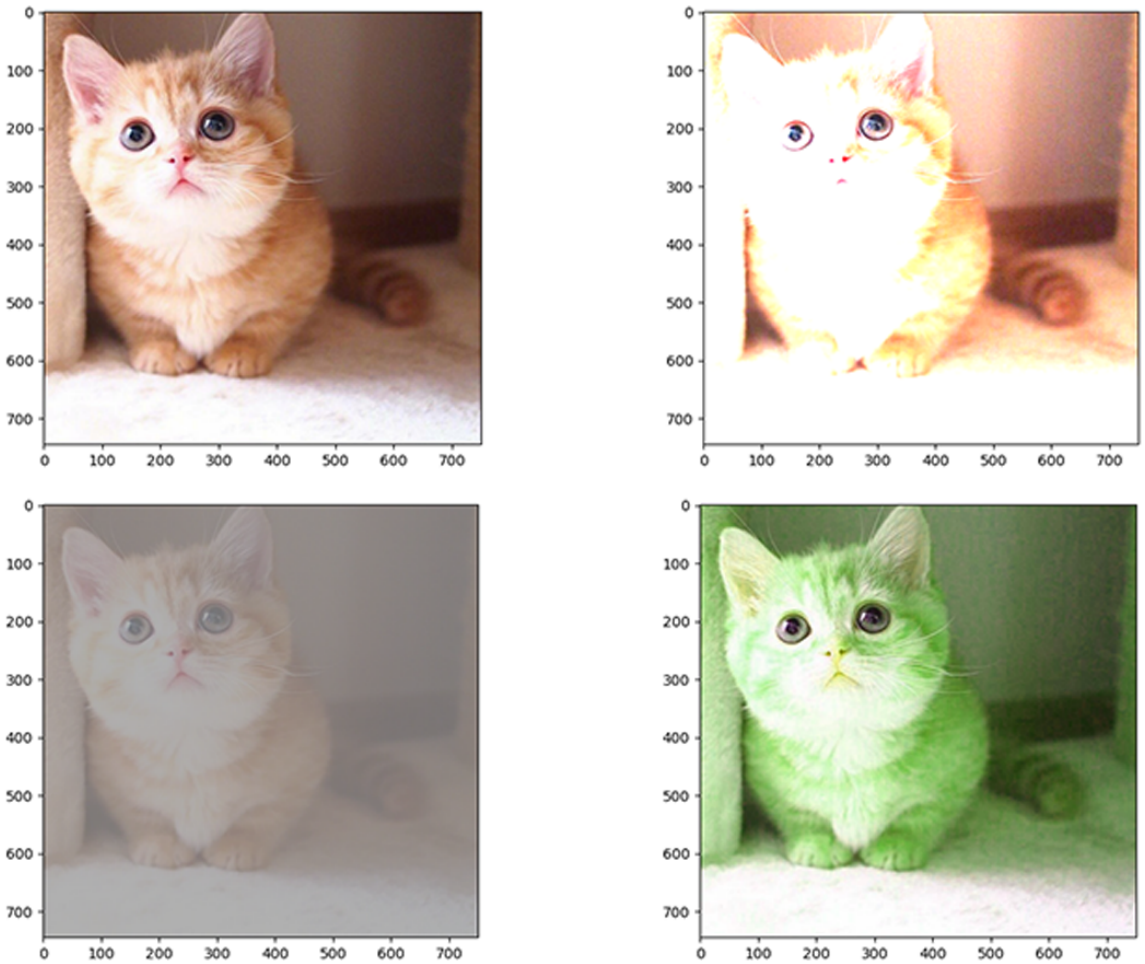


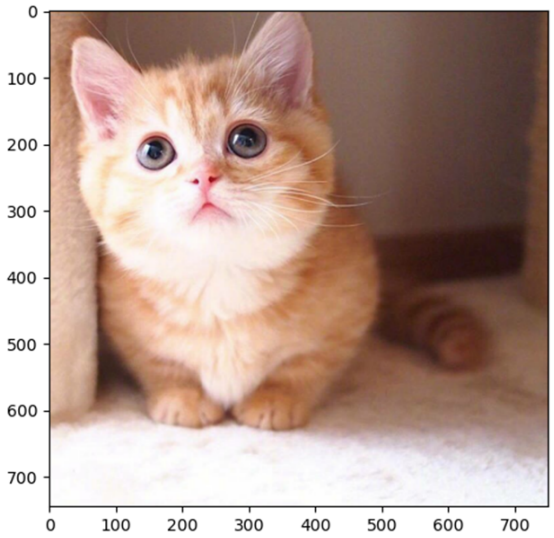
图: 亮度、对比度和颜色的随机变换

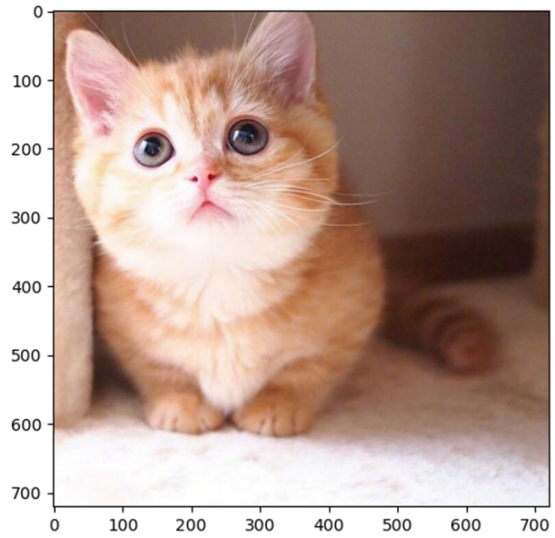
数据增强

实际运用中，图像变换方法并不是独立使用的。多个不同的图像变换可以进行自由组合，比如先随机截取，然后做翻转，再做对比度变化等等。

根据最终图像变换的强度，我们可以将数据增强分为以下三种程度：

- 原始数据（Original）：不对原始图像做形状变换，只做一些便于输入网络进行训练和测试的变换，如标准化和转换为 Tensor 等。

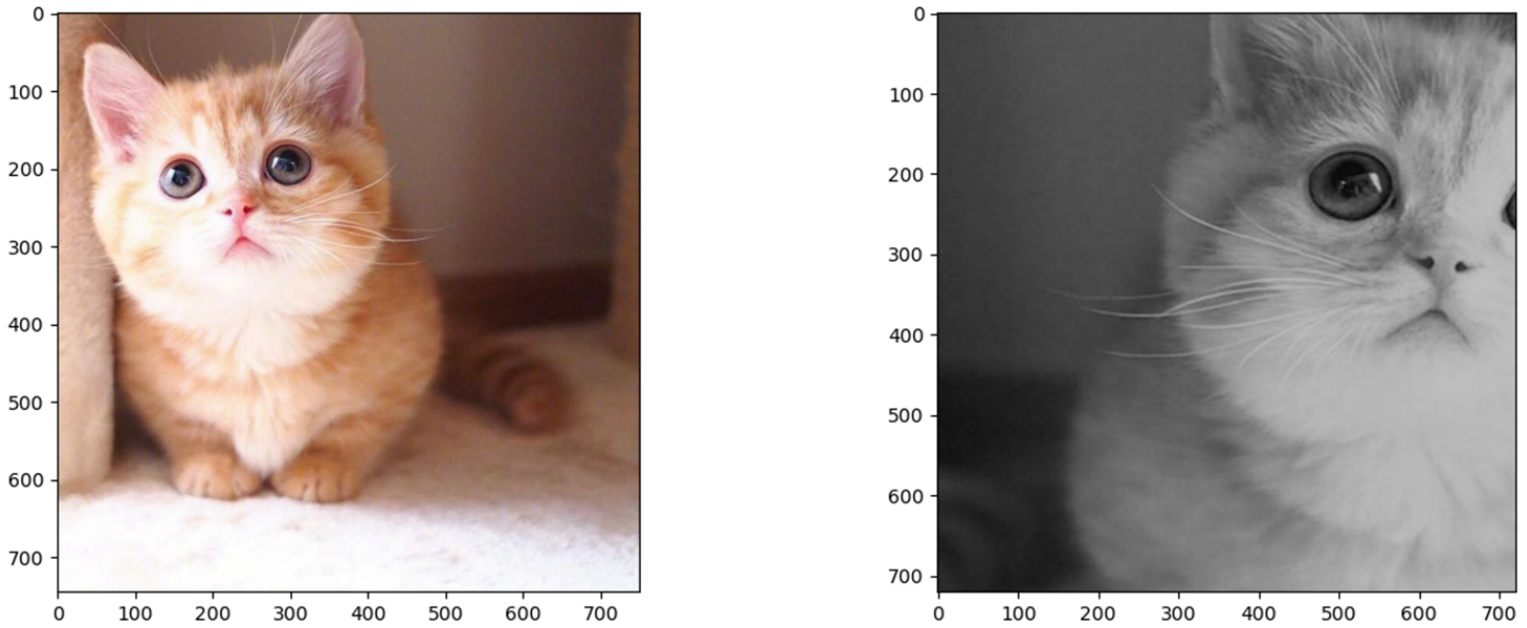




图：Original 增强

数据增强

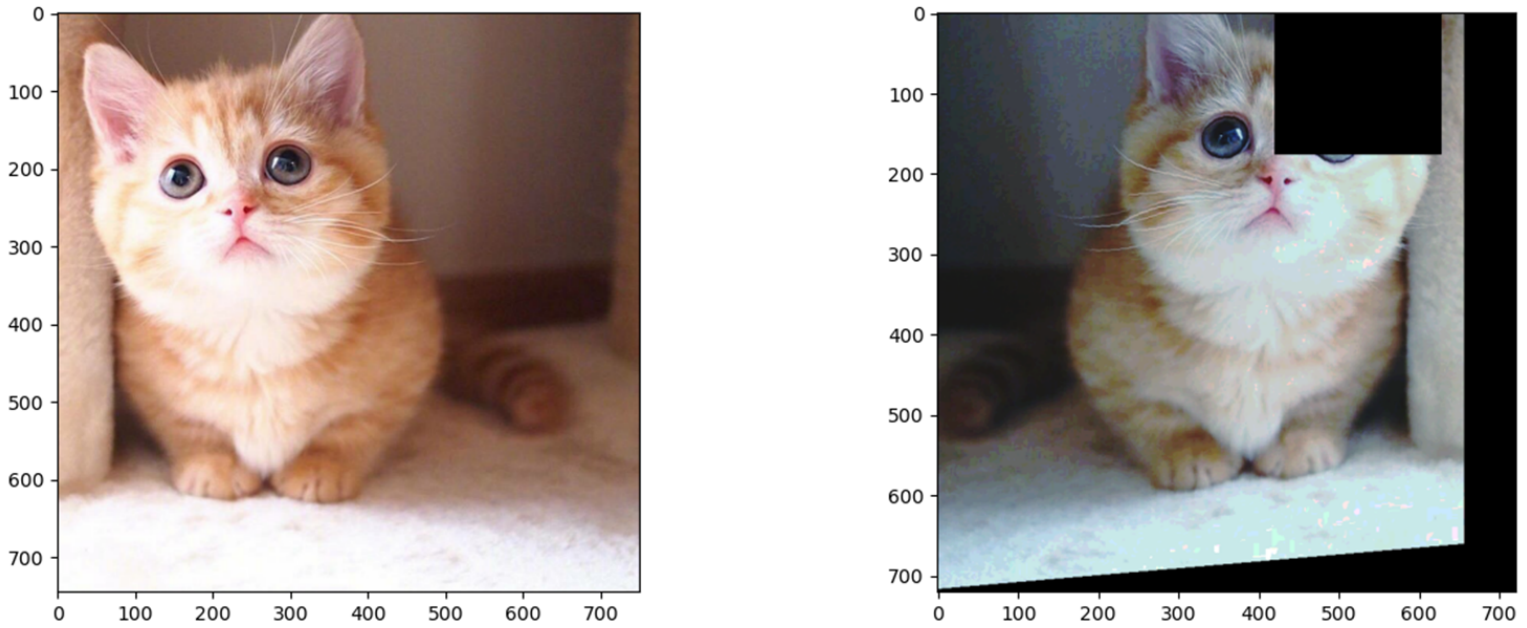
- 弱数据增强（Weak）：对图像进行较弱的数据增强，包含随机长宽比裁剪、随机水平翻转、随机颜色变化和随机灰度变化，最后进行格式转换和标准化。



图：Weak 增强

数据增强

- 强数据增强（Strong）：对图像进行较强的数据增强，在随机裁剪、随机水平翻转的基础上再施加八种随机的图像变换方法，最后进行格式转换和标准化。



图：Strong 增强

SimCLR

SimCLR（A Simple Framework for Contrastive Learning of Visual Representations，2020）是最经典的对比学习方法。它基于一个思想来构造正负样本对：对同一样本从不同视角观察到的语义都应该是相同的。

由此，SimCLR 对输入的图像进行数据增强，来模拟不同视角下的观察结果。

SimCLR 的框架中包含两个对称的分支（branch），每个分支表示一个数据增强的视图，它将同一样本的两个数据增强视图视为正样本对，不同样本之间都视为负样本对。

SimCLR

下图是 SimCLR 的网络框架，其中 Encoder 部分常用 ResNet，而 Projector 则是一个 MLP（Multi-layer Perceptron），能够显著提升效果。

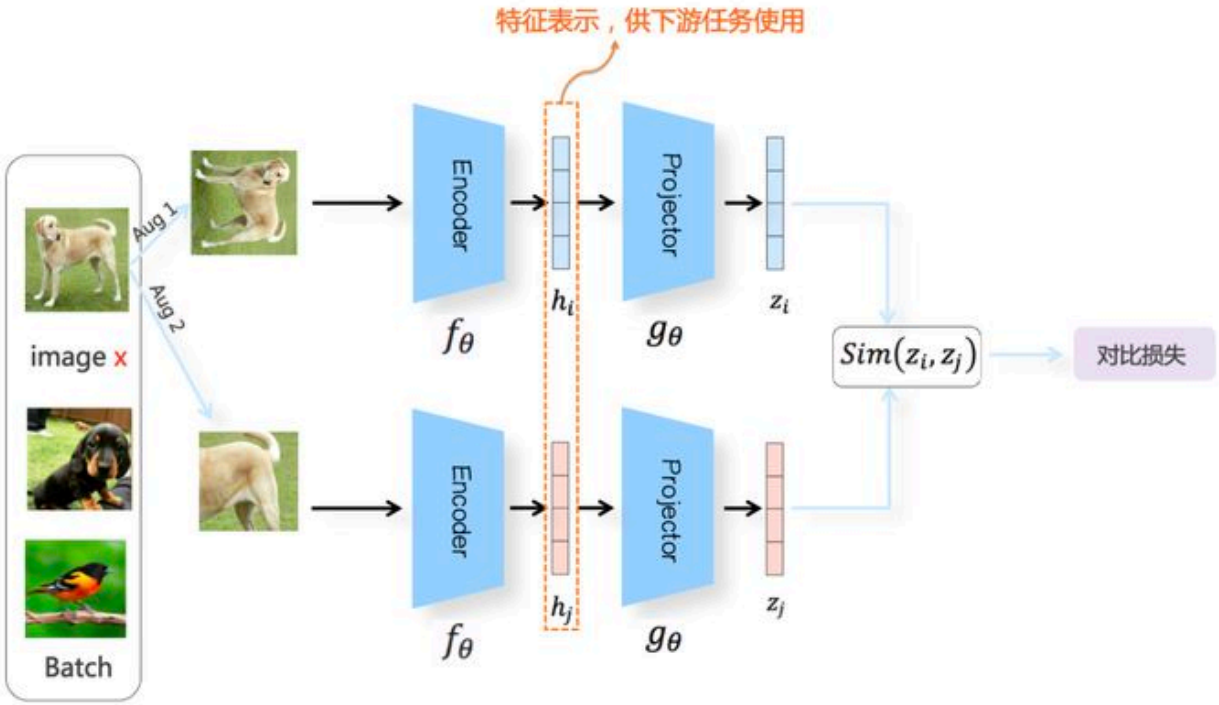


图: SimCLR 框架

SimCLR

SimCLR 使用余弦距离来衡量样本之间的对相似度:

$$S(z_i, z_j) = \frac{z_i^T z_j}{\|z_i\|_2 \|z_j\|_2}$$

对于样本 i，相对应的 InfoNCE Loss 为:

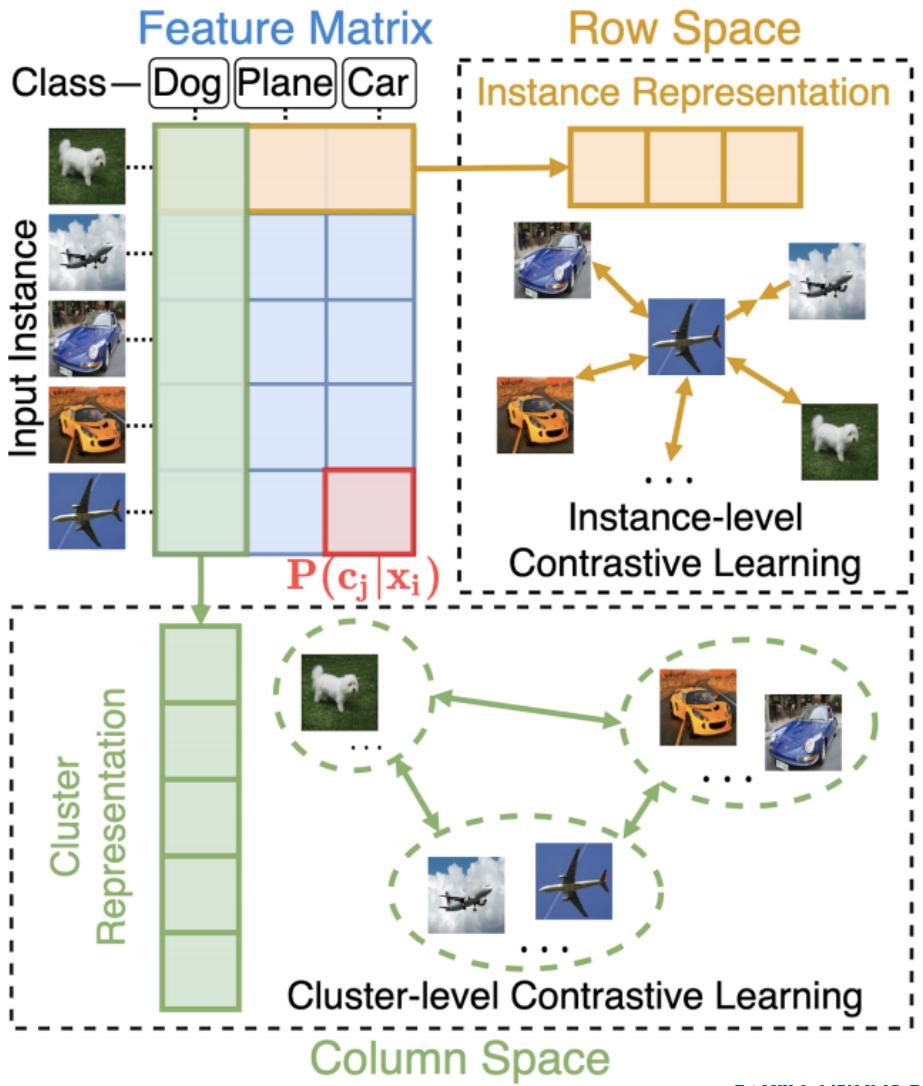
$$L_i = -\log \frac{\exp(S(z_i, z_i^+)/\tau)}{\sum_{j=1}^{N} \exp(S(z_i, z_j)/\tau)}$$

其中 τ 为温度参数，用于控制整体分布的均匀程度。

分子部分鼓励正样本对间相似度越大越好，分母部分鼓励负样本对之间相似度越小越好。

CC

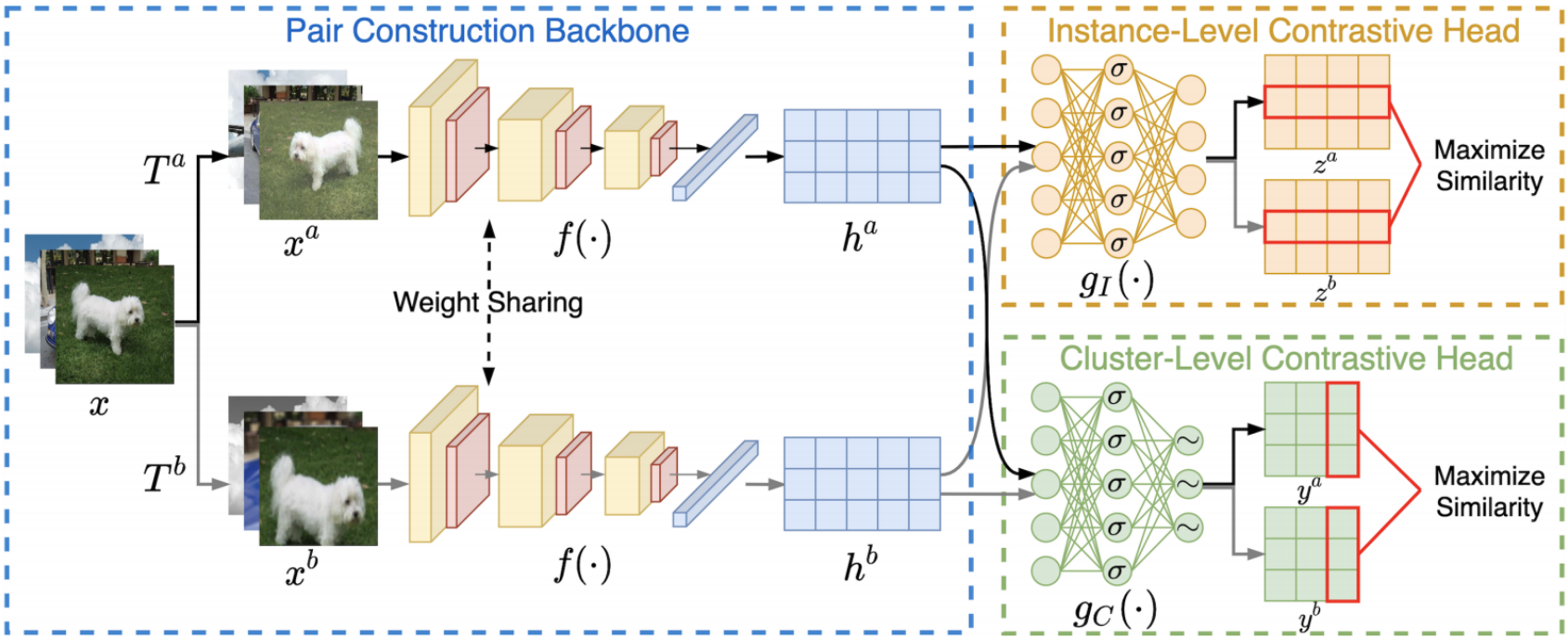
CC（Contrastive Clustering）[LHL+21] 是第一个将对比学习与聚类高度结合的工作。它以单阶段、端到端方式联合学习嵌入表示和集群分配，显式地执行实例级和聚类级的对比学习。



图：CC 框架

CC

具体而言，CC 首先通过数据增强构建数据对，学习相对应的样本特征矩阵和聚类分配矩阵。然后，分别在特征矩阵的行空间和分配矩阵的列空间中进行实例级和聚类级的对比学习。



图：CC 网络结构

CC——实例级对比学习

在实例级对比学习中，CC 对特征矩阵的行向量 z 进行对比学习。

首先定义相似度为 cosine 距离：

$$s(z_i^{k_1}, z_j^{k_2}) = \frac{(z_i^{k_1})(z_j^{k_2})^T}{\|z_i^{k_1}\|\|z_j^{k_2}\|}$$

其中 $k_1, k_2 \in \{a, b\}$ 表示 a 分支和 b 分支，$i, j \in [1, N]$。

定义同一样本 x_i 的不同视图 $\{z_i^a, z_i^b\}$ 为正样本对，其余的均为负样本对。

CC——实例级对比学习

对于一个 a 分支的样本 x_i^a，它的特征向量为 z_i^a，对应的实例级对比 Loss 为：

$$\ell_i^a = -log\frac{\exp(s(z_i^a, z_i^b)/\tau_I)}{\sum_{j=1}^{N}\left[\exp(s(z_i^a, z_j^a)/\tau_I) + \exp(s(z_i^a, z_j^b)/\tau_I)\right]}$$

其中 τ_I 为实例级的温度参数。

分子部分为正样本对，而分母部分则分别是 a 分支中的负样本和 b 分支中的负样本。

CC——实例级对比学习

总的实例级对比学习 Loss 需要考虑两个分支的所有样本，即：

$$\mathcal{L}_{ins} = \frac{1}{2N}\sum_{i=1}^{N}(\ell_i^a + \ell_i^b)$$

CC——聚类级对比学习

聚类级的对比学习则是需要我们稍微转换一下常规思想。

对于一个 batch 的样本，我们可以得到 a 分支下的分配矩阵 $Y^a \in \mathcal{R}^{N \times M}$，其中 N 为 batch 的大小，M 为簇的数量。

我们从聚类的角度来看这个分配矩阵：共有 M 列，每一列 $y_i^a (i = 1, 2, ..., M)$ 可以看做为一个类的特征。

CC——聚类级对比学习

由此，相应的正样本对即为 $\{y_i^a, y_i^b\}$，其余都为负样本对。

同样的，CC 用 cosine 距离定义类对的相似度：

$$s(y_i^{k_1}, y_j^{k_2}) = \frac{(y_i^{k_1})^T (y_j^{k_2})}{\|y_i^{k_1}\| \|y_j^{k_2}\|}$$

其中 $k_1, k_2 \in \{a, b\}$，$i, j \in [1, M]$。

CC——聚类级对比学习

对于 a 分支下的类 i，它的特征为 y_i^a，对应的聚类级对比 Loss 为：

$$\hat{\ell}_i^a = -\log \frac{\exp(s(y_i^a, y_i^b)/\tau_C)}{\sum_{j=1}^{M} \left[\exp(s(y_i^a, y_j^a)/\tau_C) + \exp(s(y_i^a, y_j^b)/\tau_C)\right]}$$

其中 τ_C 为聚类级温度参数。

CC——目标函数

同样，总的聚类级对比 Loss 需要考虑两个分支内所有的类：

$$\mathcal{L}_{clu} = \frac{1}{2M}\sum_{i=1}^{M}(\hat{\ell}_i^a + \hat{\ell}_i^b) - H(Y)$$

其中，$H(Y)$ 为 Y 矩阵的负熵，这一部分能够帮助避免平凡解。

$$H(Y) = \sum_{i=1}^{M}\left[P(y_i^a)\log(P(y_i^a)) + P(y_i^b)\log(P(y_i^b))\right]$$

$$P(y_i^k) = \sum_{j=1}^{N}\frac{Y_{ji}^k}{\|Y\|_1}, \quad k \in \{a, b\}$$

CC——目标函数

CC 的目标函数为实例级和聚类级对比 Loss 的相加。由此，CC 将表征学习和聚类学习统一在了一个单阶段端到端的框架中。

$$\mathcal{L} = \mathcal{L}_{ins} + \mathcal{L}_{clu}$$

一般还可以在两者间乘上一个权重参数，不过实验显示单纯的相加效果就已经挺好了。

CC——实验结果

Dataset	CIFAR-10			CIFAR-100			STL-10		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	0.087	0.229	0.049	0.084	0.130	0.028	0.125	0.192	0.061
SC	0.103	0.247	0.085	0.090	0.136	0.022	0.098	0.159	0.048
AC	0.105	0.228	0.065	0.098	0.138	0.034	0.239	0.332	0.140
NMF	0.081	0.190	0.034	0.079	0.118	0.026	0.096	0.180	0.046
AE	0.239	0.314	0.169	0.100	0.165	0.048	0.250	0.303	0.161
DAE	0.251	0.297	0.163	0.111	0.151	0.046	0.224	0.302	0.152
DCGAN	0.265	0.315	0.176	0.120	0.151	0.045	0.210	0.298	0.139
DeCNN	0.240	0.282	0.174	0.092	0.133	0.038	0.227	0.299	0.162
VAE	0.245	0.291	0.167	0.108	0.152	0.040	0.200	0.282	0.146
JULE	0.192	0.272	0.138	0.103	0.137	0.033	0.182	0.277	0.164
DEC	0.257	0.301	0.161	0.136	0.185	0.050	0.276	0.359	0.186
DAC	0.396	0.522	0.306	0.185	0.238	0.088	0.366	0.470	0.257
ADC	-	0.325	-	-	0.160	-	-	0.530	-
DDC	0.424	0.524	0.329	-	-	-	0.371	0.489	0.267
DCCM	0.496	0.623	0.408	0.285	0.327	0.173	0.376	0.482	0.262
IIC	-	0.617	-	-	0.257	-	-	0.610	-
PICA	0.591	0.696	0.512	0.310	0.337	0.171	0.611	0.713	0.531
CC(Ours)	**0.705**	**0.790**	**0.637**	**0.431**	**0.429**	**0.266**	**0.764**	**0.850**	**0.726**

图: CC 实验结果

CC——消融实验

CC 两个层面的对比学习影响如下，ICH 表示实例级对比学习，CCH 表示聚类级对比学习。

Dataset	Contrastive Head	NMI	ACC	ARI
CIFAR-10	ICH + CCH	**0.705**	**0.790**	**0.637**
	ICH Only	0.699	0.782	0.616
	CCH Only	0.592	0.657	0.499
ImageNet-10	ICH + CCH	**0.859**	**0.893**	**0.822**
	ICH Only	0.838	0.888	0.780
	CCH Only	0.850	0.892	0.816

CC——消融实验

CC 两个分支的数据增强对结果的影响如下，T^a 和 T^b 分别表示 a 分支和 b 分支的数据增强。

Dataset	Augmentation	NMI	ACC	ARI
CIFAR-10	$T^a(x) + T^b(x)$	**0.705**	**0.790**	**0.637**
	$T^a(x) + x$	0.630	0.690	0.533
	$x + x$	0.045	0.169	0.022
ImageNet-10	$T^a(x) + T^b(x)$	**0.859**	**0.893**	**0.822**
	$T^a(x) + x$	0.852	0.892	0.817
	$x + x$	0.063	0.177	0.030

CC——聚类过程

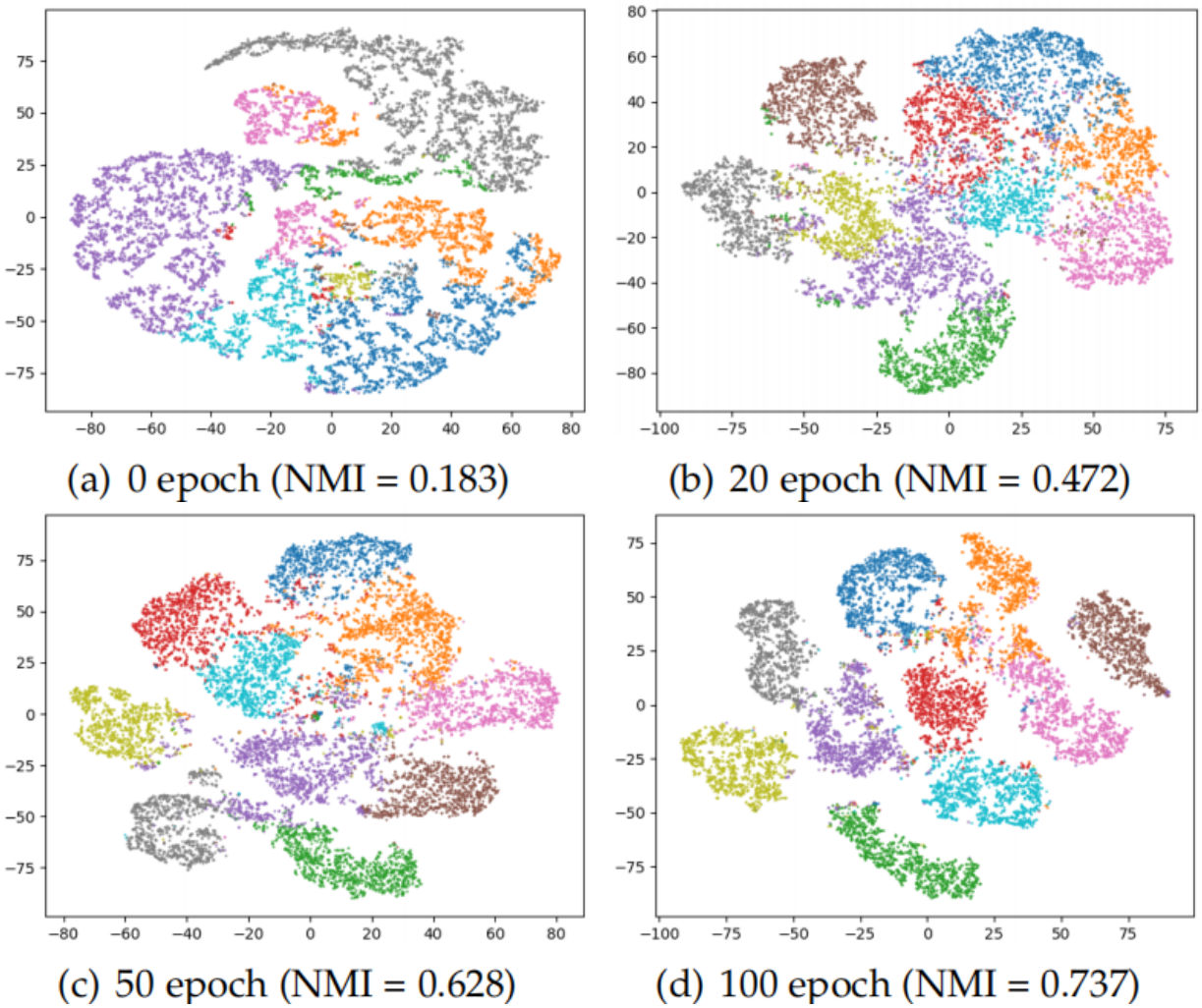


图: CC 的聚类过程

CC——方法分析

相较于其他方法，CC 提供了一种新颖的见解，即实例表示和聚类预测分别对应于可学习特征矩阵的行和列。由此，深度聚类可以优雅地统一到表示学习框架中。

CC 主要是吸收了 SimCLR 的思想，并将其运用于聚类中，没做特别多的修改。因此，还是存在一些针对聚类任务的优化空间的。而且，CC 的时间复杂度还是不低的：在单张 1080ti 上大概需要跑 70 个小时。

IDFD

IDFD（Clustering-friendly Representation Learning via Instance Discrimination and Feature Decorrelation，2020）是一个面向聚类任务的表征学习方法。

表征学习在深度聚类的效果中往往发挥着重要作用，是提升深度聚类性能的重要因素。而 IDFD 就是出于这个假设而提出的，它有两个目标：

- 学习样本间的相似性：同类样本相似度最大，不同类样本相似度最小。
- 减少特征内的相关性：特征间正交。

IDFD——主要思想

- 与 SimCLR 和 CC 一样，IDFD 将每个样本看作独立的一个类。
- 不进行数据增强，直接将一个 batch 内 n 个样本输入模型，得到一个 $n \times d$ 的特征矩阵 V。
- 样本区分：计算每个样本被分配到各个类的概率，希望分到自己对应类的概率尽可能大，其他类的概率尽可能小。
- 特征去相关：将特征矩阵 V 转置得到一个 $d \times n$ 的矩阵 V^T，每一行为一个特征维度的表示。优化目标也是使特征维度与自己相似度尽可能高，与其他特征维度相似性尽可能低。
- 聚类模块：IDFD 简单地使用了 K-means 作为聚类模块。

IDFD——方法框架

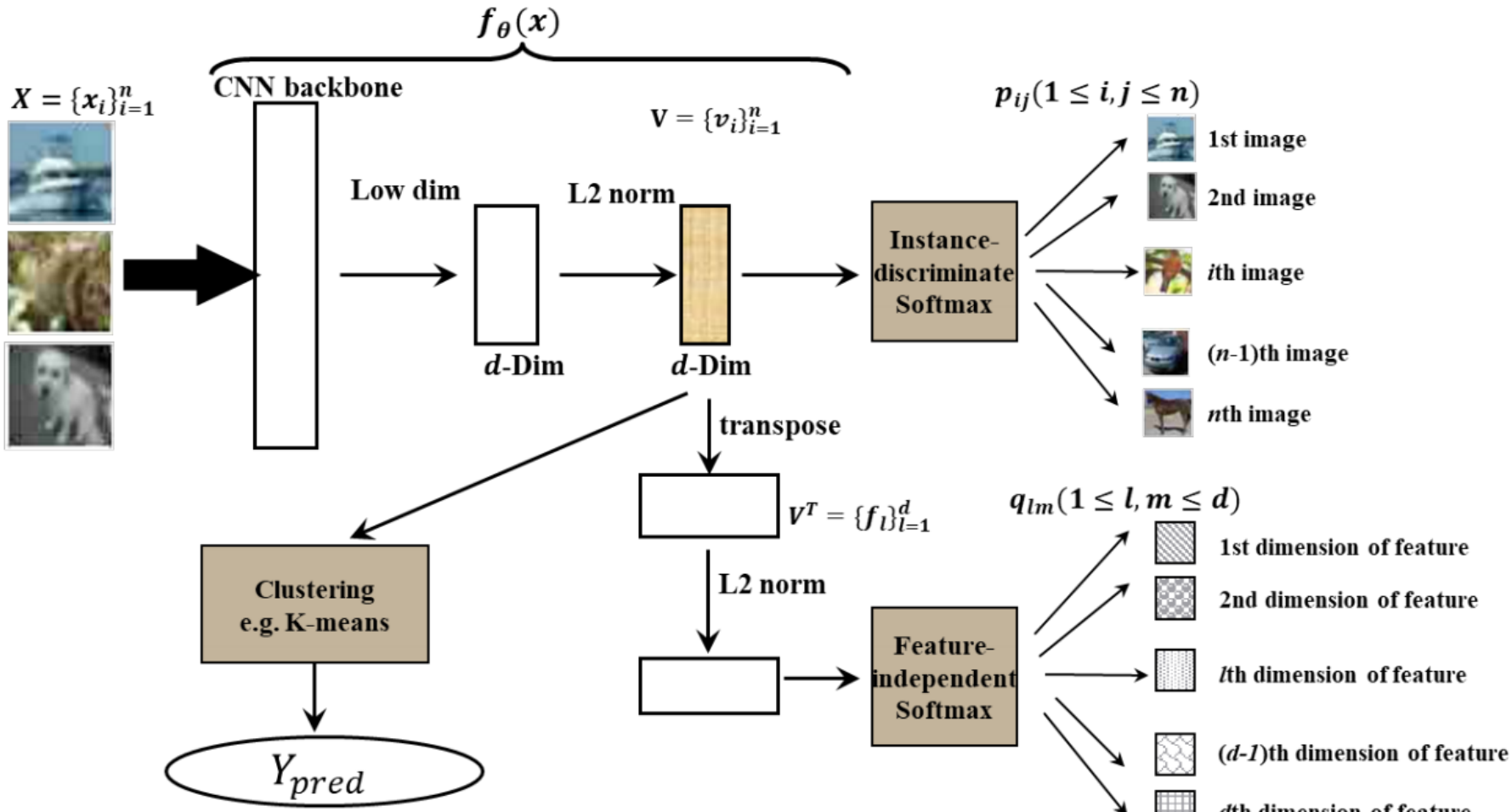


图: IDFD 框架

IDFD——样本区分

将一个 batch 内 n 个样本输入模型 f_θ，可以得到一个 $n \times d$ 的归一化后的特征矩阵 V。对于样本 x_i，它对应的表征为 $v_i = f_\theta(x_i)$。

同时，因为每个样本都对应着一个类。所以第 i 个类也可以用 v_i 来近似表示。由此，我们可以计算一个表征 v 被分到第 i 个类的概率：

$$P(i|v) = \frac{\exp(v_i^T v/\tau)}{\sum_{j=1}^{n} \exp(v_j^T v/\tau)}$$

实际上这就是一个 softmax 函数，其中 $v_j^T v$ 表示 v 和第 j 个类的匹配程度，τ 是温度参数，用来控制 softmax 的 hard 程度：τ 越大，softmax 后的结果越平滑。

IDFD——样本区分

这一部分的目标是使得联合概率 $\prod_{i=1}^{n} P_\theta(i|f_\theta(x_i))$ 最大，即：

$$L_I = -\sum_{i=1}^{n} \log P(i|f_\theta(x_i)) = -\sum_{i=1}^{n} \log(\frac{\exp(v_i^T v_i/\tau)}{\sum_{j=1}^{n} \exp(v_j^T v_i/\tau)})$$

连乘可能导致下溢，所以对其进行了对数化。

简单来说，就是对矩阵 $V^T V$ 的行空间进行了 softmax，希望最大化对角线元素。

IDFD——特征去相关

将特征矩阵 V 转置，用 f_l 表示第 l 维特征的表征，可以得到 $V^T = \{f_l\}_{l=1}^d$。

如果简单地对特征正交进行约束，那么目标可以写作：

$$L_{FO} = \|VV^T - I\|^2 = \sum_{l=1}^{d} \left((f_l^T f_l - 1)^2 + \sum_{j=1, j\neq l}^{n} (f_j^T f_j)^2 \right)$$

当 VV^T 为单位矩阵，即所有特征正交时，能够取得最小值。

IDFD——特征去相关

IDFD 还提出了一种基于 softmax 的约束：

$$Q(l|f) = \frac{\exp(f_l^T f/\tau_2)}{\sum_{j=1}^{d} \exp(f_j^T f/\tau_2)}$$

与 $P(i|v)$ 类似，$Q(l|f)$ 表示了一个特征向量与自己以及其他特征的相关性。新的特征去相关约束为：

$$L_F = -\sum_{l=1}^{d} \log Q(l|f) = \sum_{l=1}^{d} \left(-f_l^T f_l/\tau_2 + \log \sum_{j=1}^{d} \exp(f_j^T f_l/\tau_2) \right)$$

形式上与 F_I 其实是一样的。

IDFD——目标函数

为了比较各约束的效果，IDFD 定义了三种不同的目标函数：

- $L_{IDFD} = L_I + \alpha L_F$
- $L_{IDFO} = L_I + \alpha L_{FO}$
- $L_{ID} = L_I$

α 为平衡权重。

IDFD——实验结果

Dataset	CIFAR-10			CIFAR-100			STL-10		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
AE	31.4	23.9	16.9	16.5	10.0	4.8	30.3	25.0	16.1
DEC	30.1	25.7	16.1	18.5	13.6	5.0	35.9	27.6	18.6
DAC	52.2	39.6	30.6	23.8	18.5	8.8	47.0	36.6	25.7
DCCM	62.3	49.6	40.8	32.7	28.5	17.3	48.2	37.6	26.2
ID(original)	44.0	30.9	22.1	26.7	22.1	10.8	51.4	36.2	28.5
IIC	61.7	51.1	41.1	25.7	22.5	11.7	59.6	49.6	39.7
SCAN	88.3	79.7	77.2	50.7	48.6	33.3	80.9	69.8	64.6
ID(tuned)	77.6	68.2	61.6	40.9	39.2	24.3	72.6	64.0	52.6
IDFO	82.8	71.4	67.9	42.5	43.2	24.4	75.6	63.6	56.9
IDFD	81.5	71.1	66.3	42.5	42.6	26.4	75.6	64.3	57.5

图: IDFD 实验结果

IDFD——方法分析

IDFD 的思想与 CC 非常相似：

- 两者都是分别在特征矩阵的行空间和列空间上进行约束，区别在于列空间上 CC 约束的是类，而 IDFD 约束的是特征。
- 虽然 IDFD 的约束并不是显式的对比学习，但其核心思想和起到的效果还是非常类似的：最大化同类相似度，最小化异类相似度。

IDFD 的聚类方法是简单的 K-means，但还是取得了非常好的效果。这说明它学到的表征非常适合聚类，如果能配上一种更优的聚类策略和方法，效果应该能再提升一些。

SCAN

SCAN (Semantic Clustering by Adopting Nearest neighbors，2020) 采用了多阶段的架构，表征学习和聚类学习是分离的。

- 表征学习
- 聚类学习
- pseudo labeling

SCAN——表征学习

在表征学习阶段，SCAN 利用某些自监督任务作为 pretext task 来获得语义上有意义的特征。并将该步骤得到的表征作为聚类阶段的先验来初始化网络参数 Φ_θ。

这样做的好处是使得聚类学习不会依赖于低级特征。

具体实现中，SCAN 采用了 SimCLR 作为 pretext task。

SCAN——聚类学习

聚类阶段，SCAN 在特征空间 Φ_θ 中为每个图像 $x_i \in D$ 寻找最近的 k 个邻居，邻居集合表示为 $\mathcal{N}_{x_i}$。

同时，学习一个聚类分配函数 Φ_η，希望最大化 x_i 与它的 k 个邻居的聚类分配，即 $\Phi_\eta(x_i)$ 和 $\Phi_\eta(k)(k \in \mathcal{N}_{x_i})$ 之间的相似度。

注意：这里的 Φ_η 包括表征学习的部分，即 Φ_θ 初始化的参数，以及一个以 softmax 函数结束的 MLP 分类器。

SCAN——目标函数

$$\Lambda = -\frac{1}{|D|}\sum_{x\in D}\sum_{k\in\mathcal{N}_x}\log\langle\Phi_\eta(x),\Phi_\eta(k)\rangle + \lambda\sum_{c\in C}\Phi_\eta^{'c}\log\Phi_\eta^{'c}$$

其中，$\langle\cdot,\cdot\rangle$ 表示点乘操作，C 表示类的集合。

Loss 函数的第一项希望模型对样本 X 和它邻居的分配预测尽可能一致。

而第二项则是用于避免平凡解的负熵，$\Phi_\eta^{'c}$ 为：

$$\Phi_\eta^{'c} = \frac{1}{|D|}\sum_{x\in D}\Phi_\eta^c(x)$$

其中 $\Phi_\eta^c(X)$ 表示样本 x 被分配到类 c 的概率。

SCAN——self labeling

self labeling 阶段，SCAN 根据设定的阈值为一部分高置信度图像分配伪标签 c，利用 cross-entropy 训练来训练模型。每次迭代寻找新的高置信度样本，直到满足条件的样本数量不再增加。

样本 x 的软分配 $\Phi_\eta(x)$ 为一个 $1 \times K$ 的向量，定义置信度为其最大值，即满足条件 $\max(\Phi_\eta(x)) > threshold$ 的样本为高置信度样本。

SCAN

Algorithm 1 Semantic Clustering by Adopting Nearest neighbors (SCAN)

Input: Dataset $\mathcal{D}$, Clusters $\mathcal{C}$, Task τ, Neural Nets Φ_θ and Φ_η, Neighbors $\mathcal{N}_\mathcal{D} = \{\}$.
Optimize Φ_θ with task τ. ▷ Pretext Task Step, Sec. 2.1
for $X_i \in \mathcal{D}$ **do**
    $\mathcal{N}_\mathcal{D} \leftarrow \mathcal{N}_\mathcal{D} \cup \mathcal{N}_{X_i}$, with $\mathcal{N}_{X_i} = K$ neighboring samples of $\Phi_\theta(X_i)$.
end for
while SCAN-loss decreases **do** ▷ Clustering Step, Sec. 2.2
    Update Φ_η with SCAN-loss, i.e. $\Lambda(\Phi_\eta(\mathcal{D}), \mathcal{N}_\mathcal{D}, C)$ in Eq. 2
end while
while $Len(Y)$ increases **do** ▷ Self-Labeling Step, Sec. 2.3
    Y $\leftarrow$ ($\Phi_\eta(\mathcal{D}) >$ threshold)
    Update Φ_η with cross-entropy loss, i.e. $H(\Phi_\eta(\mathcal{D}), Y)$
end while
Return: $\Phi_\eta(\mathcal{D})$ ▷ $\mathcal{D}$ is divided over C clusters

图: SCAN 算法

SCAN——实验结果

Dataset	CIFAR10			CIFAR100-20		
Metric	ACC	NMI	ARI	ACC	NMI	ARI
K-means [49]	22.9	8.7	4.9	13.0	8.4	2.8
SC [54]	24.7	10.3	8.5	13.6	9.0	2.2
Triplets [41]	20.5	–	–	9.94	–	–
JULE [53]	27.2	19.2	13.8	13.7	10.3	3.3
AEVB [25]	29.1	24.5	16.8	15.2	10.8	4.0
SAE [33]	29.7	24.7	15.6	15.7	10.9	4.4
DAE [48]	29.7	25.1	16.3	15.1	11.1	4.6
SWWAE [58]	28.4	23.3	16.4	14.7	10.3	3.9
AE [2]	31.4	23.4	16.9	16.5	10.0	4.7
GAN [39]	31.5	26.5	17.6	15.1	12.0	4.5
DEC [51]	30.1	25.7	16.1	18.5	13.6	5.0
ADC [16]	32.5	–	–	16.0	–	–
DeepCluster [4]	37.4	–	–	18.9	–	–
DAC [6]	52.2	40.0	30.1	23.8	18.5	8.8
IIC [23]	61.7	51.1	41.1	25.7	22.5	11.7
Supervised	93.8	86.2	87.0	80.0	68.0	63.2
Pretext [7] + K-means	65.9 ± 5.7	59.8 ± 2.0	50.9 ± 3.7	39.5 ± 1.9	40.2 ± 1.1	23.9 ± 1.1
SCAN* (Avg $\pm$ Std)	81.8 ± 0.3	71.2 ± 0.4	66.5 ± 0.4	42.2 ± 3.0	44.1 ± 1.0	26.7 ± 1.3
SCAN† (Avg $\pm$ Std)	87.6 ± 0.4	78.7 ± 0.5	75.8 ± 0.7	45.9 ± 2.7	46.8 ± 1.3	30.1 ± 2.1
SCAN† (Best)	**88.3**	**79.7**	**77.2**	**50.7**	**48.6**	**33.3**
SCAN† (Overcluster)	86.2 ± 0.8	77.1 ± 0.1	73.8 ± 1.4	55.1 ± 1.6	50.0 ± 1.1	35.7 ± 1.7

图: SCAN 实验结果

PCL

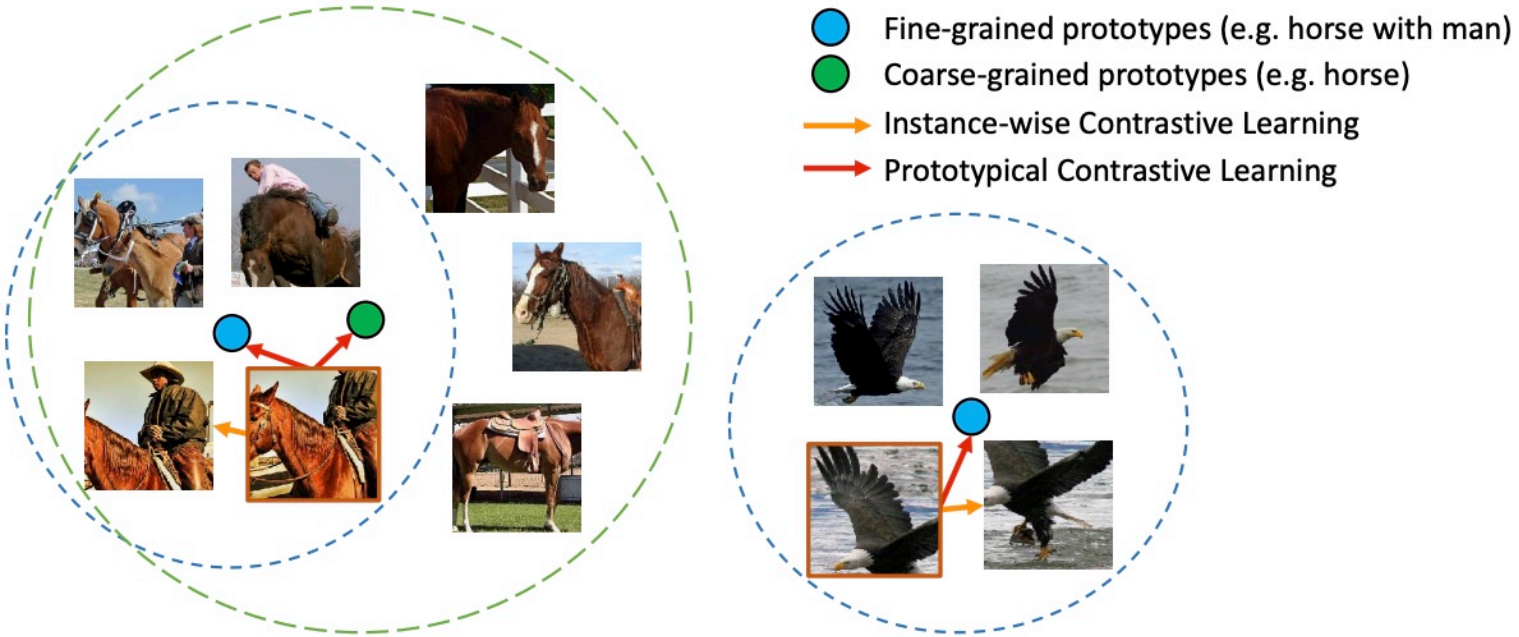
Prototypical Contrastive Learning of Unsupervised Representations (ICLR 2021)

研究动机

现有的表征学习（对比学习）主要从样本的粒度进行学习（Instance Discrimination），没有考虑全局的数据分布特性（Semantic Structure）。

现有的对比学习方法，仅能保证所有数据在高维空间中均匀分布，样本以及数据增强的样本尽量接近。

PCL


Fine-grained prototypes (e.g. horse with man)
Coarse-grained prototypes (e.g. horse)
Instance-wise Contrastive Learning
Prototypical Contrastive Learning

PCL

InfoNCE 与 ProtoNCE

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{n} -\log \frac{\exp\left(v_i \cdot v_i'/\tau\right)}{\sum_{j=0}^{r} \exp\left(v_i \cdot v_j'/\tau\right)}$$

$$\mathcal{L}_{\text{ProtoNCE}} = \sum_{i=1}^{n} -\Big(\log \frac{\exp(v_i \cdot v_i'/\tau)}{\sum_{j=0}^{r} \exp(v_i \cdot v_j'/\tau)} + \frac{1}{M}\sum_{m=1}^{M} \log \frac{\exp(v_i \cdot c_s^m/\phi_s^m)}{\sum_{j=0}^{r} \exp(v_i \cdot c_j^m/\phi_j^m)}\Big)$$

PCL

表征学习优化目标：

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n} \log p\left(x_i;\theta\right) = \arg\max_{\theta} \sum_{i=1}^{n} \log \sum_{c_i \in C} p\left(x_i, c_i;\theta\right)$$

使用 Jensen 不等式进行转化：

$$\begin{aligned}\sum_{i=1}^{n} \log \sum_{c_i \in C} p(x_i, c_i;\theta) &= \sum_{i=1}^{n} \log \sum_{c_i \in C} Q(c_i) \frac{p(x_i, c_i;\theta)}{Q(c_i)} \\ &\geq \sum_{i=1}^{n} \sum_{c_i \in C} Q(c_i) \log \frac{p(x_i, c_i;\theta)}{Q(c_i)}\end{aligned}$$

PCL

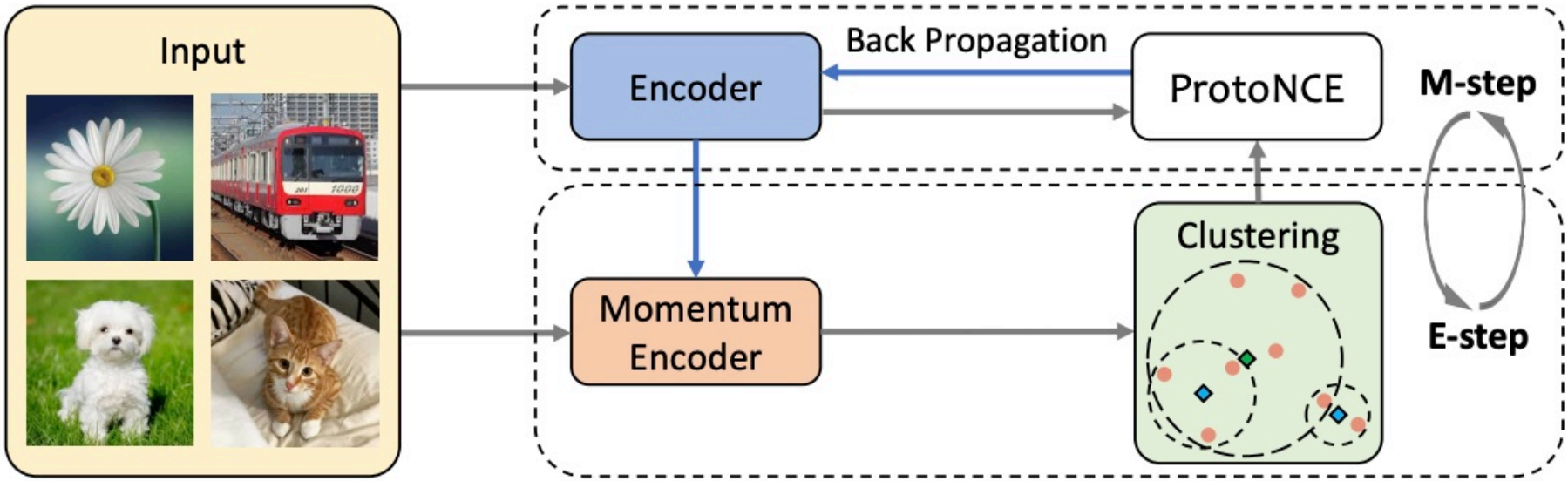


图: PCL 模型框架

PCL

PCL 的迭代优化

E 步

K-means 获得样本属于聚类的概率

M 步

$$\sum_{i=1}^{n}\sum_{c_i\in C} Q\left(c_i\right)\log p\left(x_i,c_i;\theta\right)=\sum_{i=1}^{n}\sum_{c_i\in C} p\left(c_i;x_i,\theta\right)\log p\left(x_i,c_i;\theta\right)$$

$$=\sum_{i=1}^{n}\sum_{c_i\in C} \mathbb{1}\left(x_i\in c_i\right)\log p\left(x_i,c_i;\theta\right)$$

$$p\left(x_i,c_i;\theta\right)=p\left(x_i;c_i,\theta\right)p\left(c_i;\theta\right)=\frac{1}{k}\cdot p\left(x_i;c_i,\theta\right)$$

PCL

假设每个聚类中的样本均服从高斯分布，可得

$$p\left(x_i; c_i, \theta\right) = \exp\left(\frac{-\left(v_i - c_s\right)^2}{2\sigma_s^2}\right) / \sum_{j=1}^{k} \exp\left(\frac{-\left(v_i - c_j\right)^2}{2\sigma_j^2}\right)$$

最终目标转化为

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} -\log \frac{\exp\left(v_i \cdot c_s / \phi_s\right)}{\sum_{j=1}^{k} \exp\left(v_i \cdot c_j / \phi_j\right)}$$

$$\phi = \frac{\sum_{z=1}^{Z} \left\| v_z' - c \right\|_2}{Z \log(Z + \alpha)}$$

PCL

Method	architecture (#params)	#pretrain epochs	Dataset ImageNet	VOC07	Places205
Jigsaw (Noroozi & Favaro, 2016)	R50 (24M)	90	45.7	64.5	41.2
Rotation (Gidaris et al., 2018)	R50 (24M)	–	48.9	63.9	41.4
DeepCluster (Caron et al., 2018)	VGG(15M)	100	48.4	71.9	37.9
BigBiGAN (Donahue & Simonyan, 2019)	R50 (24M)	–	56.6	–	–
InstDisc (Wu et al., 2018)	R50 (24M)	200	54.0	–	45.5
MoCo (He et al., 2020)	R50 (24M)	200	60.6	79.2*	48.9*
PCL (ours)	R50 (24M)	200	**61.5**	**82.3**	**49.2**
SimCLR (Chen et al., 2020a)	R50-MLP (28M)	200	61.9	–	–
MoCo v2 (Chen et al., 2020b)	R50-MLP (28M)	200	67.5	84.0*	50.1*
PCL v2 (ours)	R50-MLP (28M)	200	**67.6**	**85.4**	**50.3**
LocalAgg (Zhuang et al., 2019)	R50 (24M)	200	60.2†	–	50.1†
SelfLabel (Asano et al., 2020)	R50 (24M)	400	61.5	–	–
CPC (Oord et al., 2018)	R101 (28M)	–	48.7	–	–
CMC (Tian et al., 2019)	$R50_{L+ab}$ (47M)	280	64.0	–	–
PIRL (Misra & van der Maaten, 2020)	R50 (24M)	800	63.6	81.1	49.8
AMDIM (Bachman et al., 2019)	Custom (626M)	150	68.1†	–	55.0†
SimCLR (Chen et al., 2020a)	R50-MLP (28M)	1000	69.3†	80.5†	–
BYOL (Grill et al., 2020)	$R50\text{-}MLP_{big}$(35M)	1000	74.3†	-	–
SwAV (Caron et al., 2020)	R50-MLP (28M)	800	75.3†	88.9†	56.7†

其他类型数据的聚类

- 目前为止介绍的深度聚类方法都是运行在公开 CV 数据集上的算法。
- 但是可想而知，对于其他的数据类型，只要 embedding 表达的语义足够好，那么之前讲到的方法统统都适用。
- 那是不是说明聚类任务实际上退化为了表征学习呢？(是的
- 但本节讲到的其他类型数据的聚类并不是为了教大家其他类型的 embedding 怎么搞

算法的极限

- 回顾上节课学到的谱聚类，我们可以发现：谱聚类在“算法”上已经足够逼近聚类算法的极限了，那么上限已到，下限由什么决定呢？就是数据之间的关系怎么表达，以及结果好不好的标准。
- 无论是欧式聚类还是什么非线性核函数得到的距离度量，其鲁棒性还是得依靠距离度量的对象：样本特征。假如样本在特征空间中分布得足够好，理论上我们就有可能得出聚类的最优解。假如样本特征分布的足够差，那么最好的算法也无法得出最优解。
- 所以当你拿到某种数据类型的聚类对象时，你的第一反应应该是怎么无监督地学习聚类对象的表征，或者发掘聚类对象之间的关系，而不是选取某种聚类算法。

文本聚类

- 举例而言，大家在作业一中完成了邮件文本的分类器。那么拓展到一般性，普通文件怎么去聚类呢？
- 来看一个用在 Google Drive 上的文件聚类方法 [KBN+20] 的核心：

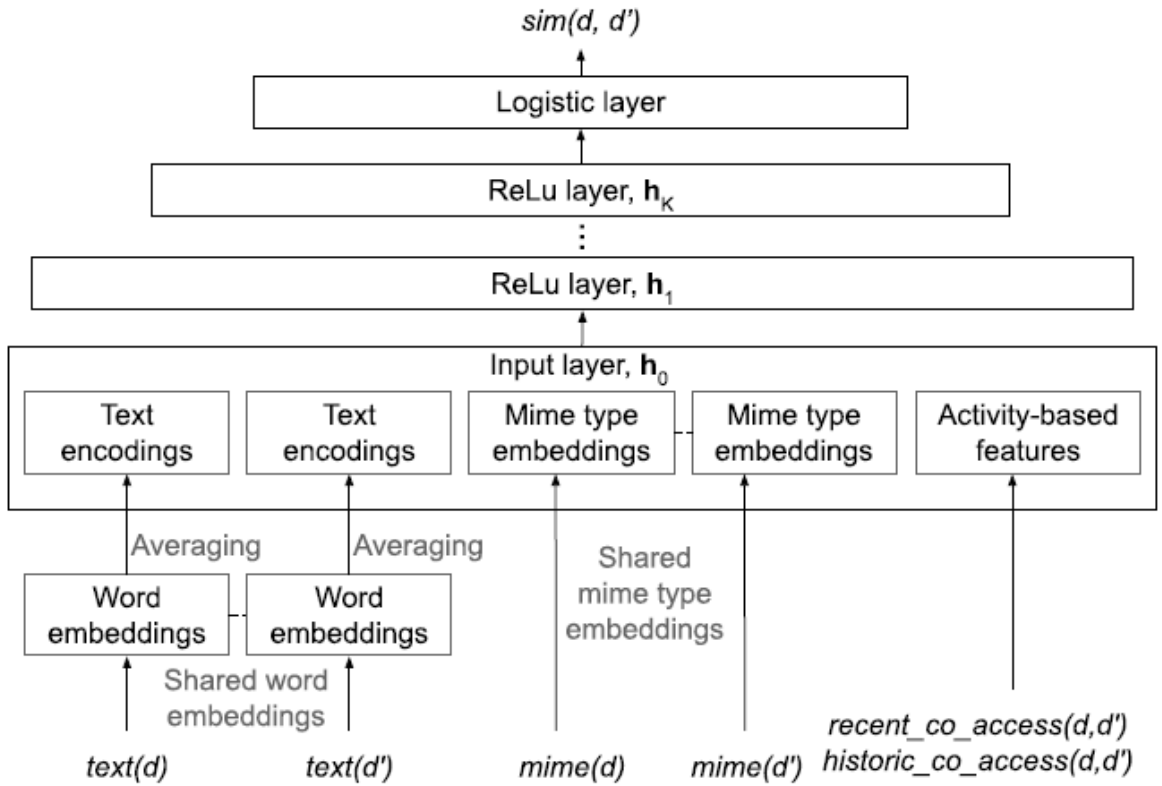
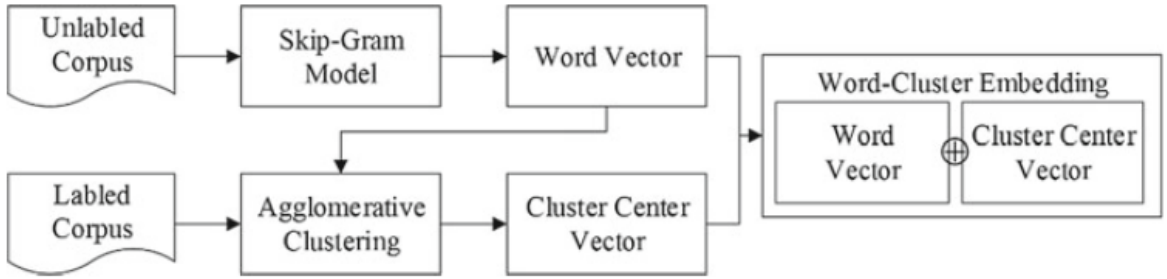


Figure 2: Document similarity model.

文本聚类

- 相类似的还有 [SZZ+18]，主题是医疗诊断文本聚类。



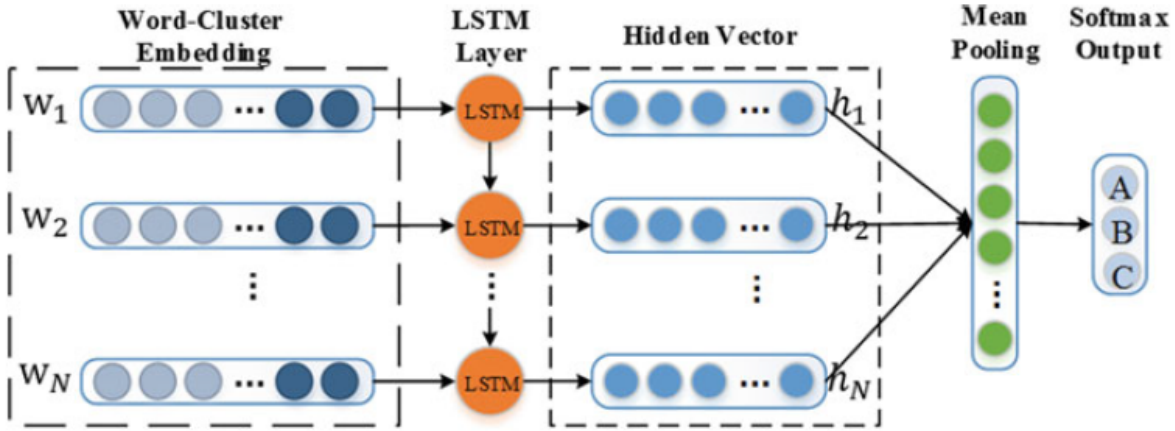


Fig. 3 The classifier using LSTM model

文本聚类

- 有的同学可能在 NLP 上有一些基础，就会想到用 BERT 系列的预训练模型提取文本类型数据的特征，之后拿去做分类或者聚类都是十分简单的问题了。
- 以此类推，各种不同的数据类型领域都有其顶尖的基于深度学习的特征工程方法。
- 那是不是说聚类只是下游任务了呢？

文本聚类–Document-Summarization

- 需要注意的是，聚类的最终目的是为了让接触数据的‘人’更方便更好地理解数据。但是聚类本身的输出是 $0-k$ 的类别标签，甚至其数值大小都不存在比较意义。
- 文本聚类的另一条研究支线被称为 Document-Summarization，翻译过来可以理解为文本摘要。
- 文本摘要是指通过各种技术，对文本或者是文本集合，抽取、总结或是精炼其中的要点信息，用以概括和展示原始文本 (集合) 的主要内容或大意。作为文本生成任务的主要方向之一，从本质上而言，这是一种信息压缩技术。聚类则是文本摘要领域中较为常用的工具。
- 通俗地讲就是从文本到聚类标签，再从聚类标签生成摘要（文本）

多模态聚类

- 上述方法无论是文本还是文件聚类，所面对的“聚类对象”都是单一的，但是对象的完整属性往往由多种模态的特征所构成。比如猫有其图像特征，有其声音特征，有其猫科动物的习性特征等等。

多模态聚类

- 举例而言 [GLWS20]Cross-Modal Subspace Clustering via Deep Canonical Correlation Analysis
- 图中甚至没有画出聚类模块，但是大家肯定猜得到聚类模块该加在哪，也能猜得到聚类模块大概是什么。

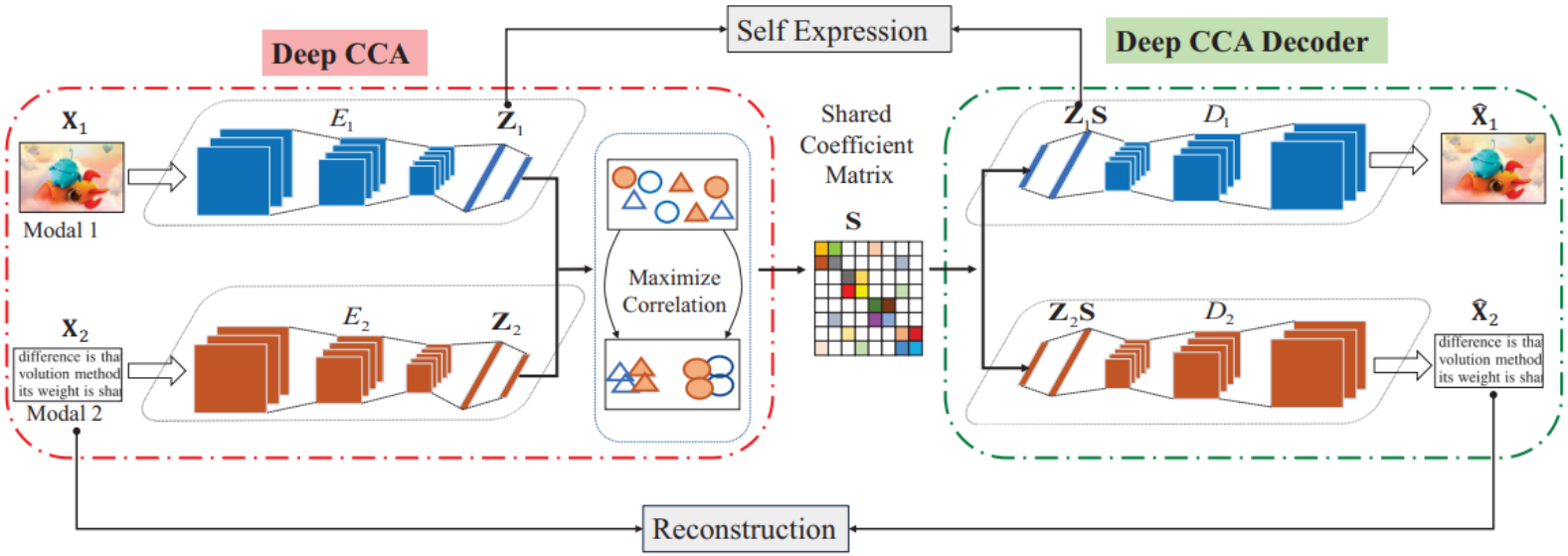


Figure 2: The framework of our proposed method(CMSC-DCCA). $\mathbf{X}_1$ and $\mathbf{X}_2$ are the input data from two modalities. $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ are the reconstruct data. E_1 and E_2 are deep convolutional encoders. $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are latent representations from the outputs of E_1 and E_2. $\mathbf{S}$ is the shared coefficient matrix. D_1 and D_2 are decoders.

多模态聚类

- 看看你的理解对不对

Algorithm 1 CMSC-DCCA

Input: Cross-modal data $\mathbf{X}_1$, $\mathbf{X}_2$; cluster number K
Output: θ, $\mathbf{S}$
Initialized: λ_1, λ_2, λ_3; learning rate $= 0.001$.
while not converge **do**
(1) Pre-train the networks using Eq.(8)
(2) Optimize network parameters $\theta_{e_1}, \theta_{e_2}$ of encoders and $\theta_{d_1}, \theta_{d_2}$ of decoders
end
while not converge **do**
(3) Train the entire networks using Eq.(9)
(4) Update parameters θ including encoders parameters $\theta_{e_1}, \theta_{e_2}$ and decoders parameters $\theta_{d_1}, \theta_{d_2}$
end
(5) Extract the self-expression coefficient matrix $\mathbf{S}$ from the training networks
(6) Compute the affinity matrix $\mathbf{C} = \frac{1}{2}(|\mathbf{S}| + |\mathbf{S}|^\top)$
(7) Perform spectral clustering on the affinity matrix $\mathbf{C}$

References I

- Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, *Deep clustering for unsupervised learning of visual features*, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.
- Q. Gao, H. Lian, Q. Wang, and G. Sun, *Cross-modal subspace clustering via deep canonical correlation analysis*, Proceedings of the AAAI Conference on Artificial Intelligence **34** (2020), no. 4, 3938–3945.
- Weize Kong, Michael Bendersky, Marc Najork, Brandon Vargo, and Mike Colagrosso, *Learning to cluster documents into workspaces using large scale activity logs*, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2416–2424.

References II

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng, *Contrastive clustering*, 2021 AAAI Conference on Artificial Intelligence (AAAI), 2021.

Ying Shen, Qiang Zhang, Jin Zhang, Jiyue Huang, Yuming Lu, and Kai Lei, *Improving medical short text classification with semantic expansion using word-cluster embedding*, International Conference on Information Science and Applications, Springer, 2018, pp. 401–411.