

- 采用这种方法后，就不会出现之前所说的，在更高层数时性能下降的问题

下面我们讨论一下，关于如何设计这个优化问题更好

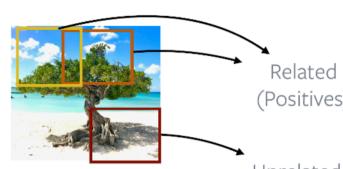
- Design Choices

1. Score Function:

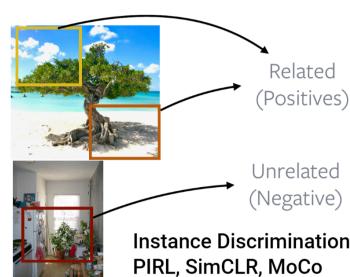
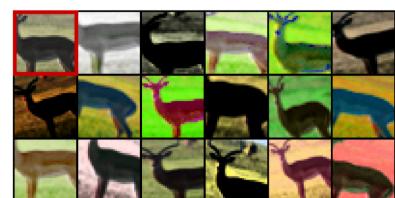
$$s(\mathbf{f}_1, \mathbf{f}_2) = \frac{\mathbf{f}_1^\top \mathbf{f}_2}{\|\mathbf{f}_1\| \|\mathbf{f}_2\|}$$

- ▶ Cosine similarity
- ▶ Commonly used

2. Examples:



3. Augmentations:

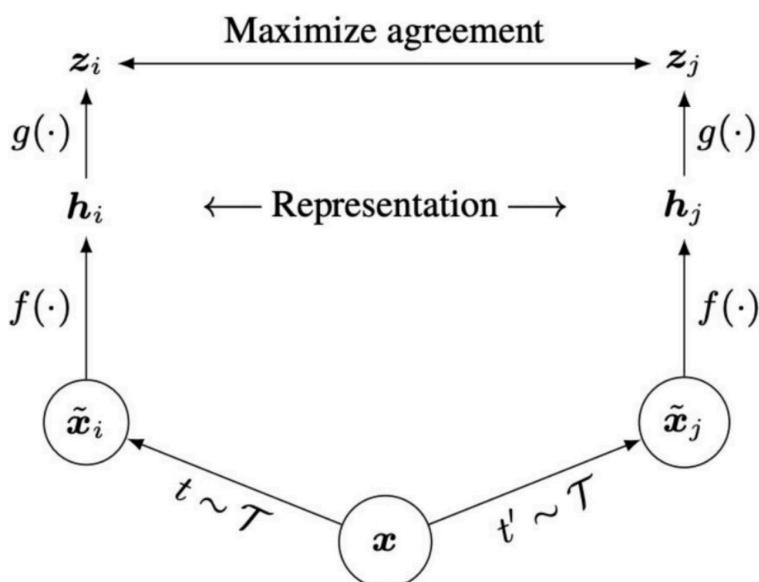


- ▶ Crop, resize, flip
- ▶ Rotation, cutout
- ▶ Color drop/jitter
- ▶ Gaussian noise/blur
- ▶ Sobel filter

- Score Function

- 关于评分函数的设计，其参数应当是 $f(x)$ ，也即我们将 views 输入到编码器中，得到该 views 的特征向量

- 正如之前所说，数学上讲，我们希望评分函数反映两个特征向量之间的相似度——最常用的方法就是余弦相似度 (Cosine similarity)——利用特征向量之间的内积除以它们的范数
- Examples
 - 我们还需要讨论如何根据参考样本取样正样本和负样本
 - Contrastive Predictive Coding
 - 这种方法中，我们从同一张图中取正样本和负样本——这种方法适用于单实例、简单场景的情况（例如图中的树），我们需要将图中描述同一个对象的部分作为相关、其余部分比如描述场景的部分作为无关
 - Instance Discrimination
 - 然而，如果一幅图只有一个对象，很可能每一块都可以是相关的——因为它们都在以不同方式描述着这个对象；此时我们就需要选择别的不同的图像作为负样本
- Augmentations
 - 关于增强图像，我们有非常多的方法——裁剪图像、缩放大小、翻转图像、旋转图像、区域裁剪、删除颜色、抖动颜色、高斯模糊、高斯噪声等
- A Simple Framework for Contrastive Learning [Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.](#)



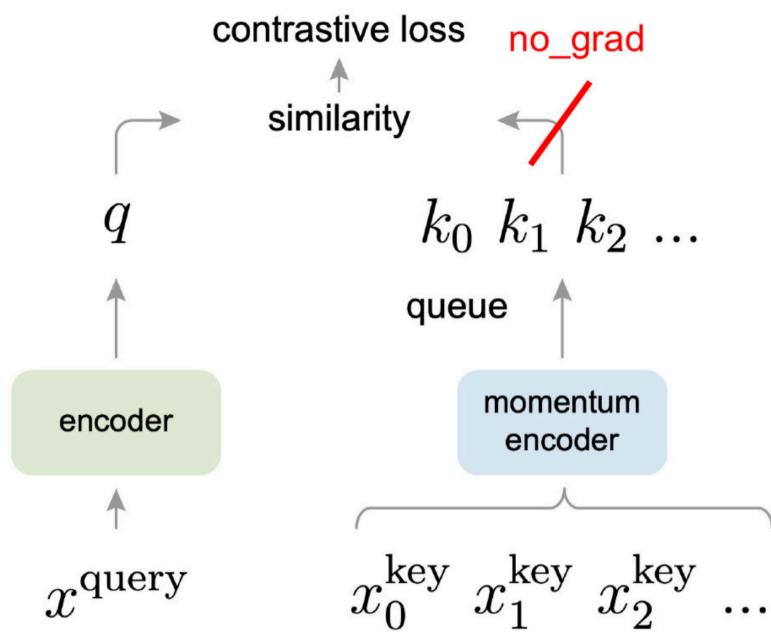
- 目前在该领域性能最好的模型之一——SimCLR
- 它使用 Cosine Similarity 作为其评分函数\$\$

$$s(\text{bf}(z_i), \text{bf}(z_j)) = \frac{\text{bf}(z_i)^T \text{bf}(z_j)}{\|\text{bf}(z_i)\| \|\text{bf}(z_j)\|}$$

- 其中它的特征向量生成过程如图所示，从 \$x\$ 取参考样本和变换后的样本，

这种对 BatchSize 的依赖使得对比学习的方法有些不切实际——这并不是我们想要的，因此提出了一种称为动量对比 (Momentum Contrast) 的方法

- Momentum Contrast (MoCo) [He, Fan, Wu, Xie and Girshick: Momentum Contrast for Unsupervised Visual Representation Learning. CVPR, 2020](#)

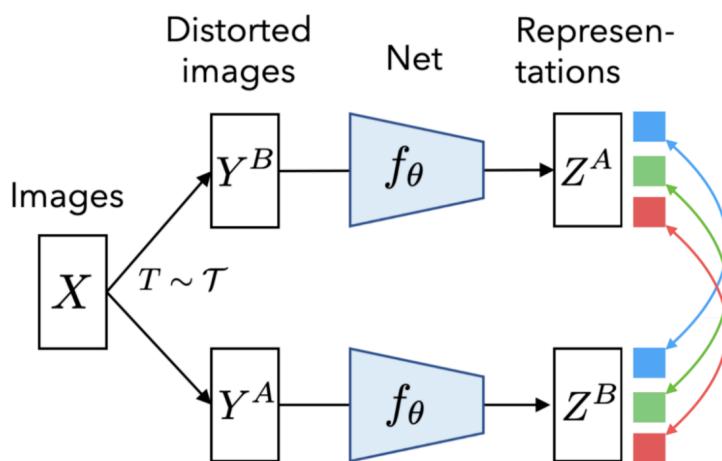


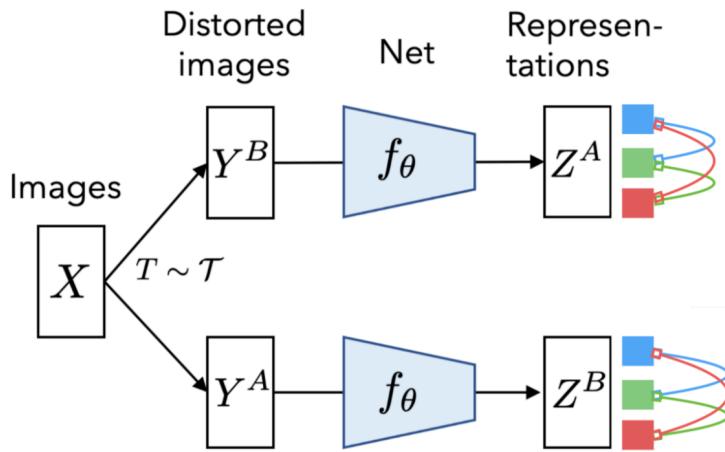
- 整体和 SimCLR 的思想类似，框架如下：
- 样本
- 类似数据库的思想，将样本分为查询样本 x^{query} （参考样本）和若干键值样本（正、负样本） x^{key}
- 编码器
- 分为查询编码器（绿色框）和动量编码器（蓝色框）
- 查询 q
- 表示当前输入样本 x^{query} 的特征
- 键值队列 k_0, k_1, k_2, \dots
- MoCo 维护一个动态更新的环形队列，用于存储负样本键值，随着新样本的加入而不断更新（想象为很多个 minibatch 组成的环形缓冲区，因此可以比固定的 batchsize 总体负样本数量更多）
- 对比损失 **contrastive loss**

- 通过计算查询 q 与键值之间的相似性得到梯度更新
- MoCo 根据对比损失，梯度的反向传播仅用于更新查询编码器，从而大大降低了计算量和内存使用
- 下面具体说说动量更新机制：
 - 由于动量编码器不会随着梯度进行更新，因此每次迭代产生的键值之间可能会存在不一致性，为了解决这一问题，MoCo 对动量编码器使用一种特定的更新机制（对相邻迭代的键值之间引入平滑约束）
 $\theta_k = \beta \theta_k + (1 - \beta) \theta_q$
 - θ_k : 动量编码器的参数 - θ_q : 查询编码器的参数 -

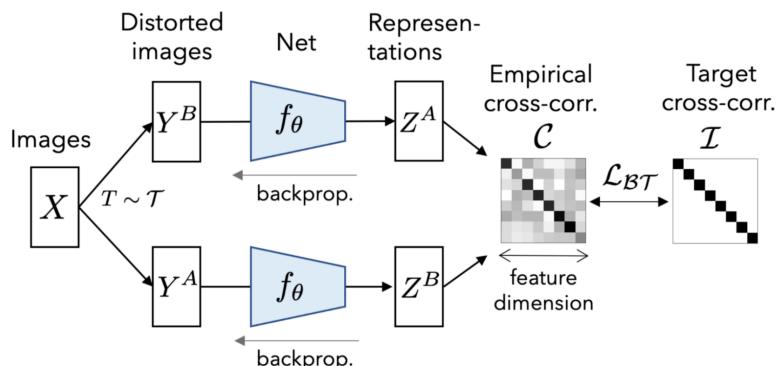
到目前为止，介绍的都是一些对比学习中经典的方法，而最近提出的一种方法称为 Barlow Twins，与这个传统架构有所不同

- Barlow Twins Zbontar, Jing, Misra, LeCun and Deny: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. Arxiv, 2021

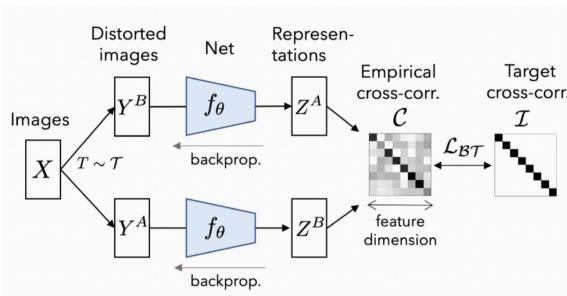




- Barlow Twins 受到信息论的启发，尝试减少神经元之间的冗余，它不再考虑图像之间的距离，而是直接考虑神经元直接的距离
- 核心思想：神经元在不同数据增强类型下生成的特征应该具有不变性
- 例如图中，假设 Z^A 和 Z^B 是 3 通道的特征向量，那么两个特征向量之间，我们希望相同颜色的的特征应该具有相似性；同时，一个特征向量的不同颜色的特征之间也应该具有差异性
- 如何实现这个思想呢——计算交叉相关矩阵 (cross-correlation matrix)，鼓励其成为为单位矩阵



- 如图所示，特征向量 Z^A 和 Z^B 的经过特征对齐后，计算得到交叉相关矩阵如图所示，——我们希望其在相同特征下（对角位置）的相关性为 1，同时不同特征下（非对角位置）的相关性为 0，也即我们鼓励交叉相关矩阵成为单位矩阵
- 具体数学计算方法如下



$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b (z_{b,i}^A)^2} \sqrt{\sum_b (z_{b,j}^B)^2}}$$

$$\mathcal{L}_{BT} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}}$$

- 交叉相关矩阵 C_{ij}

- 定义为特征维度 i 和 j 的归一化相关性 \$\$

$$C_{ij} = \frac{\sum_b z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b (z_{b,i}^A)^2} \sqrt{\sum_b (z_{b,j}^B)^2}}$$

- 损失函数 \mathcal{L}_{BT} - 由 **对角线元素一致性损失** + **非对角线元素冗余消除损

$$\mathcal{L}_{BT} = \sum_i (1 - C_{ii})^2 + \lambda \sum_{i \neq j} C_{ij}^2$$

- 由此我们可以看到 Barlow Twins 的优点——通过减少特征冗余和图像增强不变