CPC：

摘要：它介绍了一种称为 CPC 的的无监督方法，可以从高维数据中提取有用的表征，这种表征学习到了对预测未来最有用的信息。具体而言，该模型的关键是通过使用强大的自回归模型来预测潜在空间中的未来，从而学习这样的表示。作者使用一种 NCE 损失函数，该函数促使潜在空间捕获对于预测未来样本最有用的信息。它还通过使用负采样使模型变得可行。虽然以前的大多数工作都集中在评估特定模态的表示上，但作者证明了该方法能够学习有用的表示，在四个不同的领域取得了强大的性能：语音、图像、文本和强化学习。

目前无监督学习存在的问题：

- 如何更好的从原始观测值中建模高维表征，我们希望能够从数据中学习到有用的表示，这些表示可以捕捉数据的关键特征，并在后续数据中表现良好。
- 理想情况下的学习的表征到底是什么样的？是否可以使用无监督学习进行理想的表征学习？
- 作者将无监督学习与预测编码的思想相结合，提出了是否学习到一些能对未来有很好预测效果的表征？

本文的工作如下：

1. **提出 cpc 方法**：作者提出了 CPC 方法，这是一种有效的自监督学习方法，用于从高维数据中自动提取有用表示的无监督学习框架。
2. **cpc 理论**：文章阐述了 CPC 背后的原理，包括如何通过预测未来的状态来学习数据的潜在表示，并且如何利用自回归模型和概率对比损失函数来实现这一点。
3. **损失函数**：介绍了基于噪声对比估计的损失函数，这是一种用于训练模型以优化表示的方法。
4. **实验验证**：在多个不同的领域进行了广泛的实验，以验证 CPC 学习到的表示的有效性。
5. **性能比较**：将 CPC 与其他现有的无监督学习方法进行了比较，展示了其在多个任务上的优势。

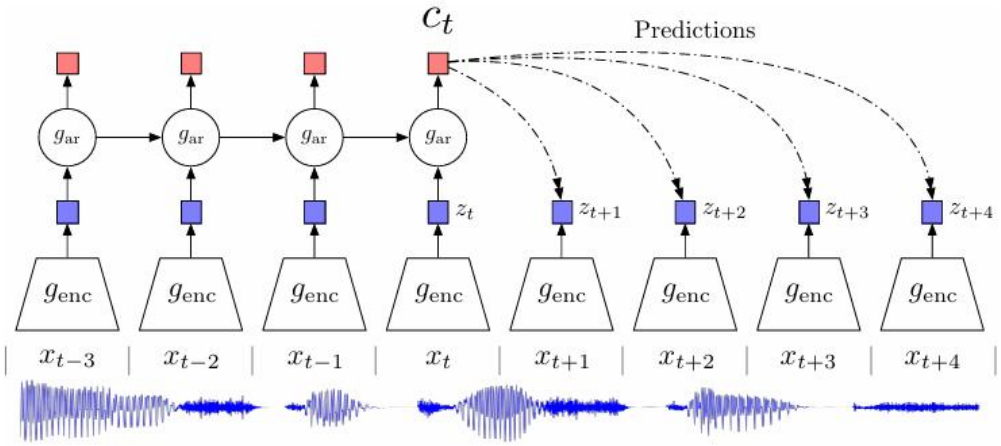
CPC 的特点：

1. **高维数据压缩**：首先，作者提出了一种将高维数据压缩成更紧凑的潜在嵌入空间的方法。在这个潜在空间中，条件预测更容易被建模。这意味着数据被转换成一个更小的格式，同时保留了对预测有用的信息。
2. **模型架构**：其次，作者在该潜在空间中使用强大的自回归模型来预测未来多个时间步的情况。自回归模型能够捕捉数据中的时间依赖性，这在处理时间序列数据（如语音或文本）时特别有用。
3. **损失函数的特点**：最后，作者采用了噪声对比估计（Noise-Contrastive Estimation, NCE）作为损失函数。（具体来说是 InfoNCE）它通过比较正样本

和负样本（噪声样本）来优化模型参数。作者指出，NCE 的这种方法与自然语言处理中用于学习词嵌入的模型相似。

4. **多模态应用**：作者将 CPC 模型应用于多种不同的数据模态，包括图像、语音、自然语言和强化学习。他们展示了 CPC 机制能够在这些领域中学习到高级信息，并且在多个任务中优于其他方法。

具体实现：



上图就是 CPC 的架构图（以语音信号为例），首先用非线性编码器 g_{enc} （如 AutoEncoder 或者 CNN） 将分割的时间窗口上的每个观测值 x_t 的进行 $z_t = g_{enc}(x_t)$ 映射，得到到 representation vector，然后再将 z_t 以及潜空间中之前所有时刻的相关信息输入到一个自回归模型 g_{ar} 中，生成当前时刻的上下文表示 $c_t = g_{ar}(z_{\leq t})$

如果要用当前的 c 去预测 k 个时刻后的 z_{t+k} ，之前提到作者不采用生成模型 $p(x_{t+k}|c)$ 进行预测，而是最大化 x_{t+k} 和 c 之间的互信息使得预测的 $\hat{z}_{t+i}$ 与真实的 z_{t+i} 尽可能的相似。

在训练之后， c_t和 z_t均可以作为 x_t的特征表示，其中z_t更关注与单个时间步的信息，而c_t可以包含多个时间步的信息。

对比学习思想：

如果c_t这个上下文的特征表示足够好（真的很好的包括当前和之前的信息的话），那它应该可以做出一些合理的预测，所以可以用c_t来预测未来时刻的$\widetilde{z_{t+1}}$、$\widetilde{z_{t+2}}$…,即未来时刻的输出。正样本是未来的输入x_{t+1}、x_{t+2}…通过编码器得到的未来时刻的特征输出z_{t+1}、z_{t+2}…。负样本的的选取就很广泛了，可以选取任意时刻的输入x_i通过编码器得到的输出z_i。它应该和预测是不相似的。（用预测的代理任务来做对比学习的。）【编码器和自回归器的参数是基于 InfoNCE 训练的】

目标：

假设 X 属于原始图像的集合，x 表示某一原始图像，Z 表示编码向量的集合，z 表示某个编码向量，$p(z|x)$表示 x 所产生的编码向量的分布。那么可以用互信息表示 X，Z 的相关性：

$$I(X,Z) = \iint p(z|x)p(x)log\frac{p(z|x)}{p(z)}dxdz$$

互信息越大意味着(大部分的)$log\frac{p(z|x)}{p(z)}$应当尽量大，这意味着$p(z|x)$应当远大于$p(z)$，即对于每个 x，编码器能找出专属于 x 的那个 z，使得$p(z|x)$的概率远大于随机概率$p(z)$。

最大化互信息，因此作者提出了一个密度比的概念：

$$f_k(x_{t+k},\ c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$$

上式$f_k(x_{t+k},\ c_t)$表示上下文c_t的预测和未来真实值x_{t+k}的相似程度，$p(x_{t+t}|c_t)$表示在给定上下文 ct 的条件下，未来观测值x_{t+k}的概率，而$p(x_{t+k})$表示未来观测值的概率。

作者利用对数双线性模型，直接用线性矩阵W_1、W_k乘以c_t来作为预测值，而z_{t+k}^K为真实值，用向量的内积衡量相似度，由此得到以下函数接近密度比：

$$f_k(x_{t+k},\ c_t) = \exp\ (z_{t+k}^T W_k c_t)$$

用该函数去表示是合理的，当真实值和c_t的乘积越大，那么也对应着更大的相似性和互信息，所以这里的$f_k(x_{t+k},\ c_t)$满足正比于互信息的条件。

优化目标：

如何优化上面的密度比使之变大呢，本文利用 InfoNCE 损失优化密度比：

$$L_N = -\ E_X[log\frac{f_k(x_{t+k},\ c_t)}{\sum_{xj\in X} f_k(x_{t+k},\ c_t)}]$$

从直观上理解，最小化损失函数等于让上式分子尽可能大，这里的分子是密度比，是之前我们希望用来描述x_{t+k}, c_t的互信息的，所以优化上述损失函数就是希望互信息值变大。

上述损失是正确分类正样本的多分类交叉熵。让我们把这个损失的最优概率写为$p(d = i|X, c_t)$，其中[d=i]是指样本x_i为正样本。则该概率有如下的形式：

$$\begin{aligned} p(d=i|X,c_t) &= \frac{p(x_i|c_t)\prod_{l\neq i} p(x_l)}{\sum_{j=1}^{N} p(x_j|c_t)\prod_{l\neq j} p(x_l)} \\ &= \frac{\frac{p(x_i|c_t)}{p(x_i)}}{\sum_{j=1}^{N}\frac{p(x_j|c_t)}{p(x_j)}}. \end{aligned}$$

P(xl)为噪声分布。$p(x_i|c_t)$为正样本分布。因此最小化 CPC 损失函数，相当于最大化 log 里面的分子，即密度比，也相当于最大化$\frac{p(x_i|c_t)}{p(x_i)}$，所以前面作者定义密度比时认为正比于该项，而与选取的样本数无关。

具体推导：（我们将 X 分为正样本x_{t+k}和X_{neg}）

$$\begin{aligned} \mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\underset{X}{\mathbb{E}}\log\left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}+\sum_{x_j\in X_{\mathrm{neg}}}\frac{p(x_j|c_t)}{p(x_j)}}\right] \\ &= \underset{X}{\mathbb{E}}\log\left[1+\frac{p\left(x_{t+k}\right)}{p\left(x_{t+k}\mid c_t\right)}\sum_{x_j\in X_{\mathrm{neg}}}\frac{p\left(x_j\mid c_t\right)}{p\left(x_j\right)}\right] \\ &\approx \underset{X}{\mathbb{E}}\log\left[1+\frac{p\left(x_{t+k}\right)}{p\left(x_{t+k}\mid c_t\right)}(N-1)\underset{x_j}{\mathbb{E}}\frac{p\left(x_j\mid c_t\right)}{p\left(x_j\right)}\right] \\ &= \underset{X}{\mathbb{E}}\log\left[1+\frac{p\left(x_{t+k}\right)}{p\left(x_{t+k}\mid c_t\right)}(N-1)\right] \\ &\geq \underset{X}{\mathbb{E}}\log\left[\frac{p\left(x_{t+k}\right)}{p\left(x_{t+k}\mid c_t\right)}N\right] \\ &= -I\left(x_{t+k},c_t\right)+\log(N) \end{aligned}$$

所以最终得到：

$$I(x_{t+k},\ c_t) \geq \log\left(N\right) - L_N^{opt}$$

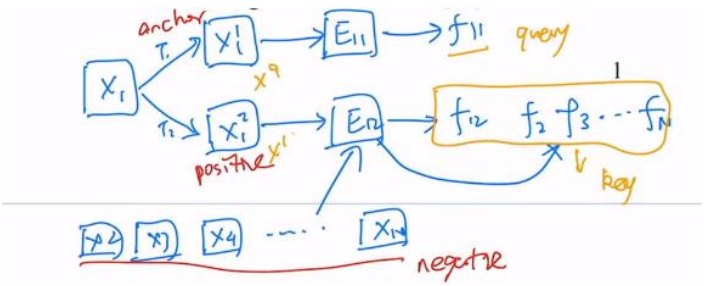
即最小化损失函数相当于最大化互信息。

实验结论部分：

作者分析了 CPC 在语音、图像、文本和强化学习等四个领域的能力，通过对比说明其取得了很强大的性能。

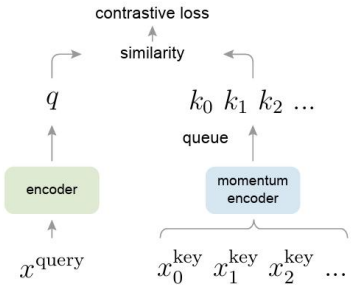
MoCo：

成果：创建了一个动态的字典，包括[1.用队列（因为队列里的样本不需要做梯度回传，所以可以向队列里放很多负样本，使字典变的很大）2.移动平均编码器（希望让字典里的特征尽量保持一致）]（如果有一个很大和比较一致性的字典则会对无监督学习有很大的好处）原作者把之前别人通过对比损失进行无监督学习都归为字典查找的问题。



比如说我们有一个图片数据集 X，我们随机选取一张图片，比如是x_1，在该图片上进行不同的变换以后得到x_1^1，x_1^2，这两张图片为一个正样本对，一般我们认为x_1^1为一个锚点(anchor)，而x_1^2为锚点对应的正样本(positive)，而数据集中的其他数据$x_2 \cdots x_N$为负样本。得到了正负样本之后我们对其进行编码，其中编码器E_{11}，E_{12}既可以是相同编码器，也可以不是(Moco 是不同的编码器)，负样本和正样本都是通过E_{12}进行编码的。具体来说，对比学习是训练一个编码器，经过编码器的输出f_{11}尽可能和与它匹配的那个 keyf_{12}特征相似，而和不匹配的进行远离。在本论文中，作者把f_{11}记为 query，$f_{12} \cdots f_N$记为 key

流程图：



创新性：

该流程图和上面的思想是类似的，但是和之前不同的是引入了队列和动量编码器。

引入队列：传统上字典的大小是 batch size 大小，由于算力的影响不可设置过大，因此很难应用大量负样本，可能会对结果造成影响。通过引入队列，在训练过程中，每一个新的 batch 完成编码后进入队列，最老的那个 batch 的 key 移除队列，这样字典的大小会与 batch size 进行剥离，这样可用的字典大小就可以大于 batch size，负样本的数目大大提升。

引入动量编码器：

$$\theta_k = m\theta_{k-1} + (1-m)\theta_q$$

MoCo 中，动量编码器是由锚点（该论文指 query）编码器初始化的，m 选取的比较大，所以动量编码器变化的比较缓慢。目的是想让队列里的特征尽可能是由相同编码器提取的，使特征保持一致性。

通过队列和引入动量编码器的设置，MoCo 可以创建一个又大又一致的字典了。

损失函数，InfoNCE：

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.

（因为类别太多，如果用 softmax 可能难以计算。NCE 就把上面的多分类问题简化为一个二分类问题，一个是数据类别，一个是噪声类别，每次拿数据样本和噪声类别做对比，但是如果把整个数据集剩下的图片作为负样本，但这样的化计算复杂度还没降下来，所以用了估计的想法，即 NCE 中的 e 代表估计的意思。即与其我把剩下的全做为负样本不如随机选取一部分再算 loss，当然负样本个数越多效果越好也即字典越大，InfoNCE 是 NCE 的变体，它认为只把任务作为二分类可能不太合理，因为负样本之间也可能有很大差距，故还是分为多分类问题比较合理，即上面公式，里面的τ是一个超参数，是控制分布的形状的，上面的 K 代表负样本的数量）。

实验部分：

method	architecture	#params (M)	accuracy (%)
Exemplar [17]	R50w3×	211	46.0 [38]
RelativePosition [13]	R50w2×	94	51.4 [38]
Jigsaw [45]	R50w2×	94	44.6 [38]
Rotation [19]	Rv50w4×	86	55.4 [38]
Colorization [64]	R101*	28	39.6 [14]
DeepCluster [3]	VGG [53]	15	48.4 [4]
BigBiGAN [16]	R50	24	56.6
	Rv50w4×	86	61.3
methods based on contrastive learning follow:			
InstDisc [61]	R50	24	54.0
LocalAgg [66]	R50	24	58.8
CPC v1 [46]	R101*	28	48.7
CPC v2 [35]	$R170^*_{wider}$	303	65.9
CMC [56]	$R50_{L+ab}$	47	64.1†
	$R50w2\times_{L+ab}$	188	68.4†
AMDIM [2]	$AMDIM_{small}$	194	63.5†
	$AMDIM_{large}$	626	68.1†
MoCo	R50	24	60.6
	RX50	46	63.9
	R50w2×	94	65.4
	R50w4×	375	**68.6**

对于在 ImageNet 的 linear classification protocol 任务，MoCo 的表现还是不错的。

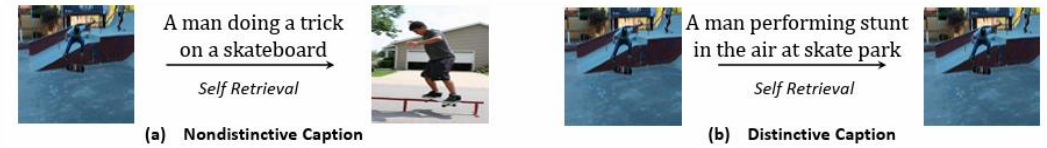
Contrastive Learning for Image Captioning

摘要部分: 图像描述生成，计算机视觉中的一个热门话题，近年来已取得了实质性进展。然而，以往工作中常常忽视了自然描述的独特性。独特性与标题质量密切相关，因为独特的标题更有可能描述图像的独特方面。在这项工作中，我们提出了一种新的学习方

法——对比学习（Contrastive Learning, CL），用于图像描述生成。具体来说，通过在参考模型之上制定的两个约束，（**正样本对的约束**：这个约束要求目标模型对于图像和其对应描述的正样本对给出更高的概率。这意味着，如果有一个图像 I 和一个正确的描述 c，目标模型 pm（$c \mid I$) 在给定图像 I 的条件下，为描述 c 赋予的概率应该高于参考模型 pn（$c \mid I$)。**负样本对的约束**：与正样本对的约束相反，这个约束要求目标模型对于图像和非对应描述的负样本对给出更低的概率。这里的负样本对是指描述 $c_{/}$是关于另一张图像的，目标模型在给定图像 I 的条件下，为这个不匹配的描述 $c_{/}$赋予的概率应该低于参考模型。）所提出的方法可以鼓励独特性，同时保持生成标题的整体质量。我们在两个具有挑战性的数据集上测试了我们的方法，它通过显著的优势改进了 baseline 模型。我们还在研究中展示了所提出的方法的通用性，它可以用于具有各种结构的模型。

面临的问题和作者的解决方案：

Introduction 部分：作者介绍到随着深度学习的进步，图像描述这一方面取得了很大的进展，但是即使是最先进的模型也有改进的空间。因为和人类的描述相比，机器生成的描述相当死板，尤其是对于那些属于同一类的图片，所生成的图片描述都非常相似，而且并没有描述出这些图片在其他方面的差异。作者把上述问题归结为独特性问题，因为当人们描述一张图像时，他们经常提及或甚至强调图像的独特方面，这些方面使它与其他图像区分开来。而机器可能会忽视这一属性。作者认为机器之所以会缺乏独特性是和模型的学习方式有关的，因为大多数图像描述的模型是用 MLE 学习的，而这种学习方式可能没有明确的考虑不同图像描述上的差异。所以作者提出了对比学习的方法，具体来说，它使用一个 baseline 模型，例如一个比较先进的模型，作为参考。在学习过程中，除了真实的图像描述对（I, c），这种方法还输入不匹配的对（I, c/），其中 c/是对另一张图像的描述。然后，目标模型被训练以满足两个目标，即（1）对于正样本对，给出比参考模型更高的条件概率 p(c|I)；以及（2）对于负样本对，给出比参考模型更低的条件概率 p(c/|I)。前者确保目标模型的整体性能不逊于参考模型；而后者鼓励独特性。作者还说这个方法是通用的，可以很好的推广到其他公式的模型。



作者通过上图解释了一下有独特性和非独特性的表示，（a）是非独特性的表示，它只指出一个男人在玩滑板特技，而对背景信息没有过多的介绍。（b）描述为一个男人在公园玩滑板特技，对描述更加准确。

当然了，在这个实验中，作者尝试给定其模型生成的描述来检索原始图像，并研究 top-k 召回率，其结果如下表所示：

	Self Retrieval Top-K Recall				Captioning	
Method	1	5	50	500	ROUGE_L	CIDEr
Neuraltalk2 [8]	0.02	0.32	3.02	27.50	0.652	0.827
AdaptiveAttention [15]	0.10	0.96	11.76	78.46	0.689	1.004
AdaptiveAttention + CL	0.32	1.18	11.84	80.96	0.695	1.029

在这个实验中，作者比较了三种不同的模型，包括通过 MLE 学习到的 Neuraltalk2 和 AdaptiveAttention，以及通过 CL 方法学习到的 AdaptiveAttention。Top-k 召回率在表 1 中列出，以及这些模型在 Rouge 和 Cider 等经典描述指标方面的整体性能。这些结果清楚地表明，自我检索的召回率与图像描述模型在经典描述指标中的性能呈正相关。

4. ContrastiveLearningforImageCaptioning 部分

在这项工作中，作者的想法是引入一个 baseline 模型作为参考，并尝试在保持生成描述的整体质量的同时增强其独特性。具体来说，引入 baseline 模型做为参考的实现主要通过以下步骤：

1. **选择 baseline 模型**：首先，选择一个已经训练好的、性能良好的图像描述生成模型作为基线模型。这个模型通常是一个最先进的模型，能够生成质量较高的描述。
2. **定义正负样本对**：在训练过程中，定义两组样本对：
    - **正样本对**：由图像及其对应的正确描述组成。
    - **负样本对**：由图像和描述另一张图像的描述组成。
3. **对比学习目标**：目标模型（即正在训练的模型）需要满足以下两个目标：
    - 对于正样本对，目标模型给出的条件概率 $pm\ (c \mid I)$ 应该大于 baseline 模型的 $)pn\ (c \mid I)$。
    - 对于负样本对，目标模型给出的条件概率 $pm\ (c_{/} \mid I)$ 应该小于 baseline 模型的 $pn\ (c_{/} \mid I)$。
4. **损失函数**：设计一个损失函数，它能够反映上述两个目标。损失函数通常包括两部分，一部分鼓励正样本对的概率高于 baseline 模型，另一部分惩罚负样本对的概率低于 baseline 模型。
5. **训练过程**：在训练过程中，目标模型通过最小化损失函数来学习。这个过程涉及到最大化正样本对的概率差异和最小化负样本对的概率差异，从而使得目标模型在生成描述时能够更加关注图像的独特特征。

4.2 对比学习

在对比学习（CL）中，我们通过相对于一个参考模型$p_n(:;\varphi)$的行为约束来学习目标图像描述生成模型$p_m(:;\theta)$。学习过程需要两组数据：(1) 观测数据 X，这是一组真实的图像-描述对（$(c_1, I_1), (c_2, I_2), ..., (c_{Tm}, I_{Tm})$），在任何图像描述生成数据集中都容易获得；(2) 噪声集 Y，包含不匹配的对（$(c_{/1}, I_{/1}), (c_{/2}, I_{/2}), ..., (c_{/Tn}, I_{/Tn})$），可以通过为每个图像$I_t$随机采样$c_{/t} \in C_{/IT}$生成，其中$C_{/IT}$是除图像$I_t$的所有真实描述之外的所有真实描述的集合。我们称 X 为正样本对，而 Y 为负样本对。

对于任何一对(c, I)，目标模型和参考模型将分别给出它们估计的条件概率$p_m(c|I;\theta)$和$p_n(c|I;\varphi)$。我们希望对于任何正样本对(c_t, I_t)，$p_m(c|I;\theta)$大于$p_m(c|I;\varphi)$，反之亦然对于任何负样本对$(c_{/t}, I_t)$。按照这个直觉，我们最初的尝试是定义 D((c, I); θ, φ)，即$p_m(c|I;\theta)$和$p_n(c|I;\varphi)$之间的差异，为 $D((c,I);\theta,\varphi)= p_m(c|I;\theta) - p_n(c|I;\varphi)$,并将损失函数设置为：$L'(\theta; X, Y, \varphi) = \sum_{t=1}^{Tm} D((c_t, I_t); \theta, \varphi) \quad - \sum_{t=1}^{Tn} \quad D((c_{/t}, I_t); \theta, \varphi).$

在实践中，这种公式会遇到几个困难。首先，$p_m(c|I;\theta)$和$p_n(c|I;\varphi)$非常小（约 1e-8），这可能导致数值问题。因此分布对其取对数。其次，由于负样本是随机采样的，不同的正负样本所产生的 D((c, I); θ, φ)大小也不一样，有些 D 可能远远大于 0，有些 D 则比较小，而在最大化 loss 的过程中更新较小的 D 则更加有效，因此作者使用了一个逻辑回归 。最后重新定义损失函数为：

$$L(\theta; X, Y, \varphi) = \sum_{t=1}^{Tm} \ln\left(h\left((c_t, I_t); \theta, \varphi\right)\right) \quad - \sum_{t=1}^{Tn} \quad ln(1 - h((c_{/t}, I_t); \theta, \varphi))..[9]$$

$$h\left((c_t, I_t); \theta, \varphi\right) = r_\gamma\left(G\left((c_t, I_t); \theta, \varphi\right)\right) = \frac{1}{1 + \gamma \exp\left(-G\left((c_t, I_t); \theta, \varphi\right)\right)}$$

$$G\left((c_t, I_t); \theta, \varphi\right) = lnp_m(c|I;\theta) - lnp_n(c|I;\varphi)$$

对于ν = Tn/Tm 的设置，我们选择ν = 1，即$Tn = Tm$，以确保正面和负面对的平衡影响。这种设置在我们的实验中始终产生良好的性能。此外，我们复制 X K 次，并采样 K 个

不同的 Y，以便在不过度拟合的情况下涉及更多样化的负对。在实践中，我们发现 K = 5 足以使学习稳定。最后，我们的目标函数定义为：$J(\theta) = \frac{1}{K} \quad \frac{1}{Tm}\sum L(\theta; X, Y_k \quad , \varphi)$.

4.3 讨论

最大似然估计（MLE）是图像描述生成领域的流行学习方法。MLE 的目标是仅最大化真实图像描述对的概率，这可能会导致一些问题，包括生成描述之间的高度相似性。而在 CL 中，真实对的概率通过正面约束（方程(9)中的第一项）间接确保，而负面约束（方程(9)中的第二项）抑制了不匹配对的概率，迫使目标模型也从独特性中学习。

对比学习的体现

1. **正负样本对的对比**：CL 方法通过区分正样本对（图像与其正确描述的配对）和负样本对（图像与错误描述的配对）来训练模型。这种方法鼓励模型对于正样本对赋予更高的概率，而对于负样本对赋予更低的概率，以此提升描述的区分性和独特性。
2. **目标模型与参考模型的对比**：CL 使用一个性能良好的参考模型（如最先进的模型）作为基准，目标模型在学习过程中与之进行对比。目标模型不仅需要达到与参考模型相似的整体性能，还要在生成独特描述的能力上超越参考模型。
3. **损失函数的设计**：CL 的损失函数设计考虑了正负样本对的对比。损失函数包括两部分，一部分是最大化目标模型对正样本对的条件概率，另一部分是最小化目标模型对负样本对的条件概率。这种设计促使模型在保持准确性的同时增加描述的多样性。

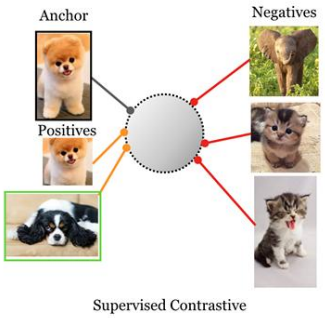
Supervised Contrastive Learning

摘要：

在这项工作中，作者扩展了自监督的小批量对比方法到完全监督的设置中，允许有效利用标签信息。通过在嵌入空间拉近同一类别的点的聚类，同时推开来自不同类别的样本聚类。并分析了两种可能的监督对比（SupCon）损失函数版本，并确定了最佳性能的损失公式。

引言部分：

作者第一段指出交叉熵损失函数可能存在的不足之处，并指出了对比学习在自监督学习的思想：在嵌入空间中拉近一个锚点和一个“正样本”，并推开许多“负样本”。由于没有标签可用，正样本通常由数据增强的样本组成，而负样本通过对小批量中的随机选择样本形成。然后又提出了一个用于监督学习的损失，它建立在对比自监督文献的基础上，通过利用标签信息。来自同一类别的归一化嵌入被拉近，而不是来自不同类别的嵌入。作者在这项工作中的技术新颖性是考虑每个锚点的多个正样本，以及多个负样本（与使用单个正样本的自监督对比学习相反）。这些正样本是从与锚点相同类别的样本中抽取的，而不是像在自监督学习中那样是锚点的数据增强。如下图所示：



它将同一批中来自同一类别的所有样本集作为正样本，与批中剩余样本作为负样本进行对比。正如黑白小狗的照片所展示的，考虑类别标签信息的结果是在嵌入空间中，同一类别的元素比自监督情况下更加紧密地对齐。

但作者认为这种方法如何正确设置损失函数并不明显，因为自监督的损失函数并不能直接使用。

$$L^{self} = \sum_{i \in I} L_i^{self} = -\sum_{i \in I} log \frac{\exp\left(z_i z_{j(i)}/\tau\right)}{\sum_{a \in A(i)} \exp\left(z_i z_a/\tau\right)}$$

其中$i \in I = \{1 \ldots 2N\}$, $A(i) = I \setminus \{i\}$,z_i是样本的向量表示，其中$j(i)$是样本 i 对应的正样本，其余 2N-2 个为负样本。从损失函数的设计来看，我们可以看出自监督学习无法处理样本带标签属于同一个类的情况，因为这时候自监督对比学习认为它们是负样本。作者在这里分析了两种替代方案。

作者的主要贡献如下：

1. 提出了对比损失函数的一个新颖扩展，允许每个锚点有多个正样本，从而将对比学习适应于完全监督的设置。
2. 展示了该损失函数在多个数据集上提供了一致的 top-1 准确率提升。
3. 分析证明了该损失函数的梯度鼓励从困难的正样本和负样本中学习。
4. 展示了与交叉熵相比该损失对一系列超参数不太敏感。

方法部分：

作者首先说明了一下模型的框架：该方法在框架结构上与用于自监督对比学习的中的方法相似，但针对监督分类进行了修改。给定一批输入数据，我们首先应用数据增强两次以获得数据的两个副本。这两个副本都通过编码器网络进行前向传播，以获得一个 2048 维的归一化嵌入。在训练期间，这种表示会进一步通过一个投影网络进行传播，该网络在推理时会被丢弃。在投影网络的输出上计算监督对比损失。为了使用训练好的模型进行分类，我们在冻结的表示上训练一个线性分类器，使用交叉熵损失。

以下是几个关键步骤：

1. **数据增强**：对输入数据批次进行两次增强，得到两个不同的副本。
2. **编码器网络**：将增强后的数据副本通过编码器网络进行前向传播，得到归一化的嵌入表示。
3. **投影网络**：在训练阶段，编码器的输出会通过投影网络进一步处理，但这个网络在最终的模型推理中不会使用。
4. **监督对比损失**：基于投影网络的输出计算监督对比损失，该损失考虑了类别标签信息，以拉近同类样本的表示，推开不同类样本的表示。
5. **分类器训练**：在训练好的嵌入表示的基础上，训练一个线性分类器，使用交叉熵损失来进行分类任务。

下面是对监督对比损失函数的介绍：

$$\mathcal{L}_{out}^{sup} = \sum_{i \in I} \mathcal{L}_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau)} \tag{2}$$

$$\mathcal{L}_{in}^{sup} = \sum_{i \in I} \mathcal{L}_{in,i}^{sup} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau)} \right\} \tag{3}$$

其中z_l是通过预测头得到的向量表示，其中$P(i) = \{p \in A(i) : y_p = y_i\}$代表了所有的正样本。$A(i) = I \setminus \{i\}$,$I$是所有样本的集合。

作者认为上述的损失函数有以下属性：

1. 可以推广到任意数量的正样本
2. 随着负样本数量的增加，对比能力增强。
3. 内在的进行硬正样本/负样本挖掘的能力。

上面的两个损失函数是不等价的，根据下图，L_{out}^{sup}更优一些。

Loss	Top-1
$\mathcal{L}_{out}^{sup}$	78.7%
$\mathcal{L}_{in}^{sup}$	67.4%

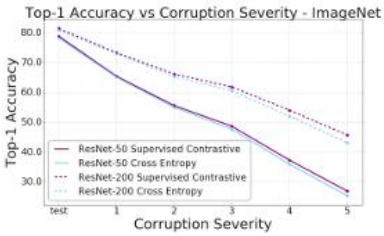
作者认为这是归一化项$\frac{1}{|P(i)|}$带来的偏差，因为在L_{in}^{sup} 中，正例归一化项位于对数内部，导致对整个损失函数的影响仅仅为常数级别。而前者在对数外部，能有效地影响梯度变化。故在以下实验中只考虑L_{out}^{sup}。
实验部分：

Dataset	SimCLR[3]	Cross-Entropy	Max-Margin [32]	SupCon
CIFAR10	93.6	95.0	92.4	**96.0**
CIFAR100	70.7	75.3	70.5	**76.5**
ImageNet	70.2	78.2	78.0	**78.7**

Loss	Architecture	Augmentation	Top-1	Top-5
Cross-Entropy (baseline)	ResNet-50	MixUp [61]	77.4	93.6
Cross-Entropy (baseline)	ResNet-50	CutMix [60]	78.6	94.1
Cross-Entropy (baseline)	ResNet-50	AutoAugment [5]	78.2	92.9
Cross-Entropy (our impl.)	ResNet-50	AutoAugment [30]	77.6	95.3
SupCon	ResNet-50	AutoAugment [5]	**78.7**	**94.3**
Cross-Entropy (baseline)	ResNet-200	AutoAugment [5]	80.6	95.3
Cross-Entropy (our impl.)	ResNet-200	Stacked RandAugment [49]	80.9	95.2
SupCon	ResNet-200	Stacked RandAugment [49]	**81.4**	**95.9**
SupCon	ResNet-101	Stacked RandAugment [49]	80.2	94.7

鲁棒性上：

Loss	Architecture	rel. mCE	mCE
Cross-Entropy (baselines)	AlexNet [28]	100.0	100.0
	VGG-19+BN [44]	122.9	81.6
	ResNet-18 [17]	103.9	84.7
Cross-Entropy (our implementation)	ResNet-50	96.2	68.6
	ResNet-200	69.1	52.4
Supervised Contrastive	ResNet-50	**94.6**	**67.2**
	ResNet-200	**66.5**	**50.6**



可以发现作者提出的方法相较于之前的成果还是有一定的提升的。