作者：小哲

微信公众号：小哲AI

原文来自微信公众号

github：https://github.com/lxztju

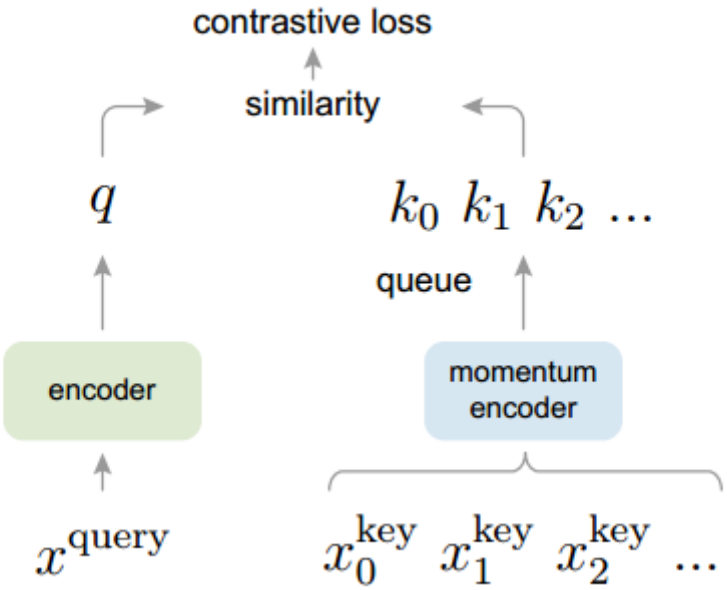
自监督学习修炼之MoCov1与MoCov2

这篇文章是何凯明大神一篇关于自监督学习的一篇文章,这篇文章的方法简单有效.

1. MoCov1

1. 论文摘要

针对无监督的视觉表示学习,我们提出了一种动量对比的方法(Momentum Contrast, MOCO)。对比学习往往被看作一个字典查找的问题，我们使用队列与移动平均编码器构建了一个动态的字典。这使我们能够动态地构建一个大而一致的词典，从而促进无监督的对比学习。MoCo在ImageNet的共用线性协议下提供了非常有竞争性的结果。更重要的是，MoCo学到的表示在下游的任务中也会迁移的非常好，MoCo在7个检测/语义分割任务（pascal voc， 吃哦吃哦， 和一些其他的数据集）上可以超过他的有监督训练版本。有时会超出很多。这表明在有监督与无监督表示学习上的差距在许多视觉任务中已经变得非常近。



2. 论文主要思想

基于对比表示的自监督学习的一般思路（*就我看过的一点点文章得到的理解认识*）就是：

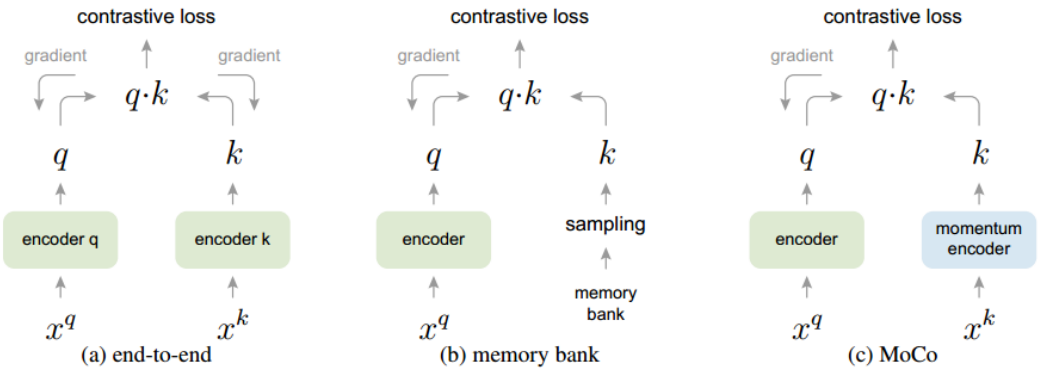
对于给定的一个样本x， 选择一个（或者一批）正样本y（这里正样本的对于图像上的理解就是x的不同的data augmentation版本）。然后选择一批负样本（对于图像来说，就是除了x之外的图像），然后设计loss function来将x与正样本之间的距离拉近，负样本之间的距离推开

这个思想有点像 **KL散度** 是吧. 看到这儿感觉思想和triplet loss的思路也蛮像的，感觉都是这样的，**同类之间距离小，不同类别之间的距离大**。

先说两个主要的词query与key，这里的理解query就是上边说的x， key就是上文中的y。这里采用键key主要就是考虑采用使得query与字典中对应的key相似，与其他的键不相似。

然后说一下本文的思想及解决的问题：

1. 传统上字典的大小就是batch-size，由于算力的影响不能设置过大，因此很难应用大量的负样本。因此效率较低。
2. 为了解决这个问题，本文采用**队列**来存储这个字典，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出队列，这样**字典的大小与batchsize实现分离**，这样可用的字典的大小就可以远远大于batchsize，负样本的数目就会大大扩增，效率得到大幅提升。



三种方案的对比：

a的方案就是传统的方案，采用batchsize作为字典的大小进行端到端的训练。

b的方案是采用一个较大的memery bank存储较大的字典（存储所有的样本），但是每次进行query之后才会对memory进行更新，所以每次采样得到的query可能是很多步骤之前编码的的向量，这样就丢失了一致性。（这里怎么理解，就是memory bank中存储了所有的样本，每次进行query时就采样其中的一部分，但是只有这次训练之后才会更新对应的key，由于每个step编码器都会进行更新，这样最新的query采样得到的key可能是好多步骤之前的编码器编码得到的key，因此丧失了一致性）

从这里来看，end-to-end的方法一致性最好，但是受限于batchsize的影响。而memory-bank的方法字典可以设置很大，但是一致性却较差，这看起来似乎是一个不可调和的矛盾。

这里kaiming大神利用momentum（移动平均更新模型权重）与queue（字典）轻松的解决这个问题。

论文中给出了算法的伪代码：

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.

使用queue，每次query之后都删除最早的batch的样本，然后将最新的batch更新入队，这样就巧妙的缓解了memory-bank一致性的问题，同时利用队列可以保存远大于batchsize的样本，这样也解决了end-to-end的batch-size的耦合问题。

为kaiming大神献上膝盖。

论文地址：[Momentum Contrast for Unsupervised Visual Representation Learning]

2. MoCov2

一篇很短的文章， 只有3页。

1. 论文摘要

对比无监督学习近些年已经获得了很大的成功，例如MoCo，SimCLR。在这篇文章（这里原文是note）中，我们整合SimCLR中的两个主要提升在方案到MoCo中，并且验证了SimCLR算法的有效性。我们使用一个MLP的projection头（SimCLR中的方案）还有更多的数据增广（也是SimCLR中的）与MoCo整合在一起，我们构建了一个效果优于SimCLR的baseline，并且不需要很大的训练batch。我们希望这种方案可以使现在最优的无监督学习方法有更大的可用性。

2. 论文主要思想

整合SimCLR到MoCo中。

两种方案的结构图。

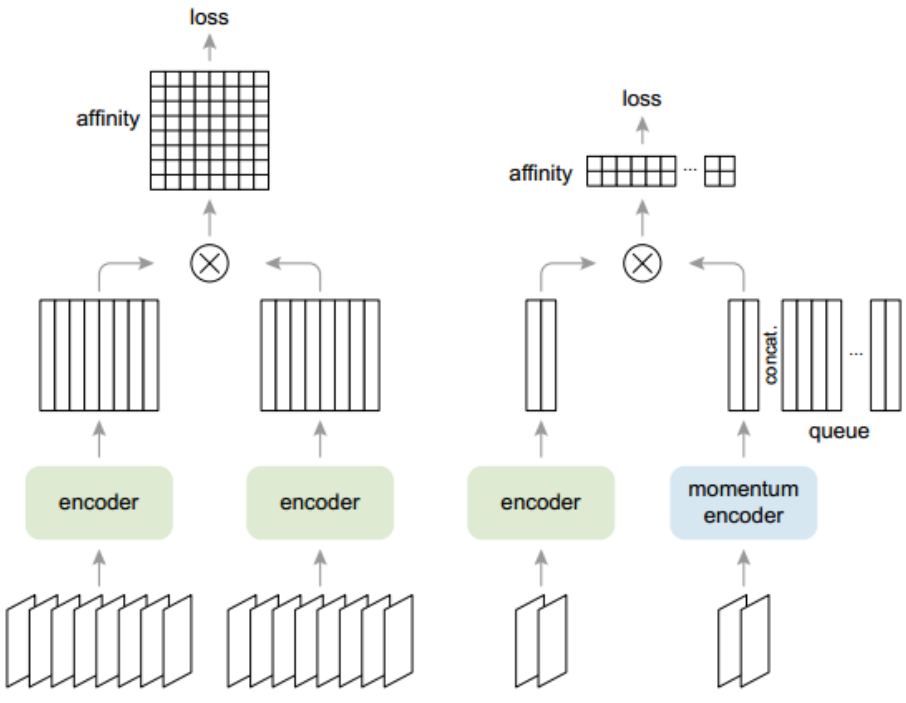


Figure 1. A **batching** perspective of two optimization mechanisms for contrastive learning. Images are encoded into a representation space, in which pairwise affinities are computed.

论文地址：[Improved Baselines with Momentum Contrastive Learning]

我自己刚开始了解这个自监督主题，没看多少论文，每看一篇文章都感叹大佬牛X，有很多理解都不到位，欢迎交流指正。