1. 不使用预训练模型：

使用监督学习，test score=0.31490

使用**pseudo label**，test score=**0.33818**

使用MixMatch，test score=0.00872

1. 使用ResNet官方预训练模型：

使用**监督学习**，test score=**0.91090**

使用pseudo label，test score=0.90290

1. 使用MoCo预训练模型+微调：

实验所用模型：ResNet50

训练方法：有监督学习、Pseudo Label

数据集组织方式：

1. 将有标注数据按二八比例划分为训练集和验证集，训练集与无标签数据混合，组成最终训练集；
2. 不设验证集，将所有的有标签数据和无标签数据用作训练集；

共可以得到4种不同的训练方式

本次实验所用的backbone模型是ResNet（18/50），在不使用预训练参数的前提下，不难看出，无论是使用哪种半监督学习机制（或者直接使用监督学习），在验证集上的分类准确率都达不到80%。然而使用了ImageNet上预训练过的ResNet后，只需10轮左右的训练，训练准确率（、验证准确率）就能够达到98%左右，由此看来预训练是十分必要的。因此想到了先用无监督方法（如MoCo）进行预训练，然后再在训练好的模型上尝试使用pseudo label、MixMatch等无监督方法进一步提升模型的泛化能力。

MoCo：

MoCo的核心思想是通过判断两个小片是否来自同一张图片来预训练一个图像分类模型。

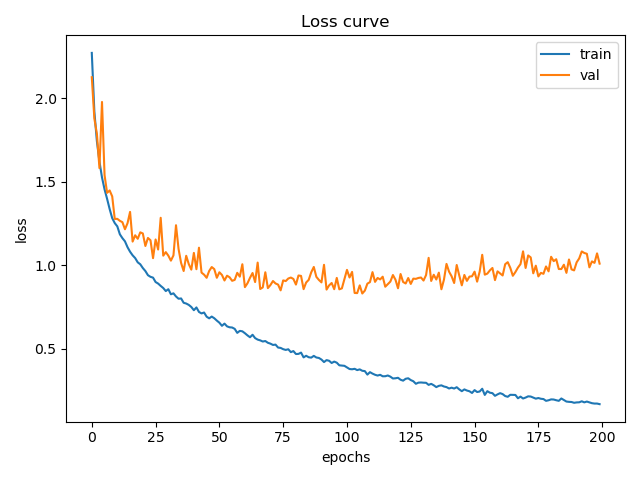
然后给定一个小片就理所当然应该有有与之对应的正例（和该小片来自同一张图片）和负例（和该小片来自不同图片），这些都是之前的对比学习中已经提出来的思想了

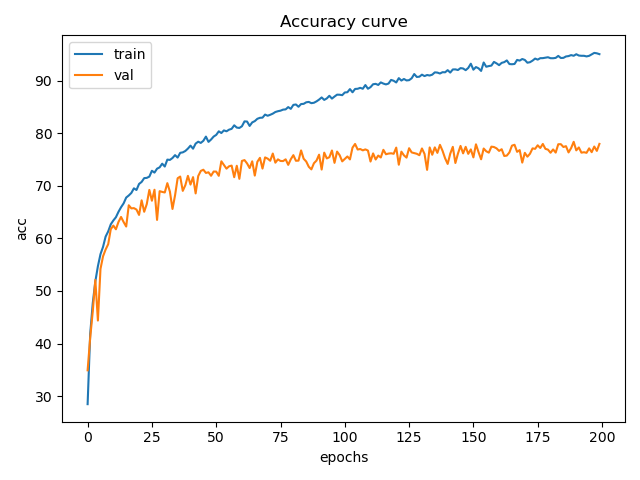
然后MoCo的不同之处在于它在每轮训练中对于一个给定的小片，给出1个正例、K个例，小片称之为query，使用query encoder编码、正例和负例称之为key，使用key encoder编码，目的就是要两个encoder分别给出适当的编码，使得模型能够从K+1个key中找到与query所对应的正例和负例，也就是训练了两个编码器，让它们分别给出query和key的合理表示，然而问题的关键在于整个模型中并不涉及分类器！但是依然可以使用相似度来度量query和任意一个key之间的相似度，只要query和正例之间的相似度高、和负例的相似度低即可！

关于MoCo的几个离奇问题：

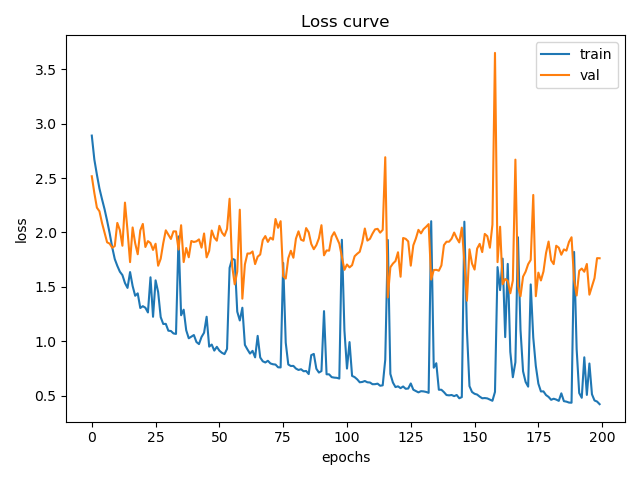
1. Query和Key（无论正例或负例）分别是按照不同的编码方式给出的（使用不同的编码器），并且key编码器更新很慢，在query编码器看来完全就没有改变，给出的不管正例还是负例的编码全都是很无脑的，根本区别不出正例key与负例key之间的差别，所以算出来query与正负例的差别都不大，导致训练没啥结果。一言以蔽之，使用两个编码器来编码query和key的训练方式本身就是有问题的；
2. MoCo预训练了两个编码器（其中个人感觉key编码器就没怎么训练），那么下流任务中微调的过程中到底用哪个呢？

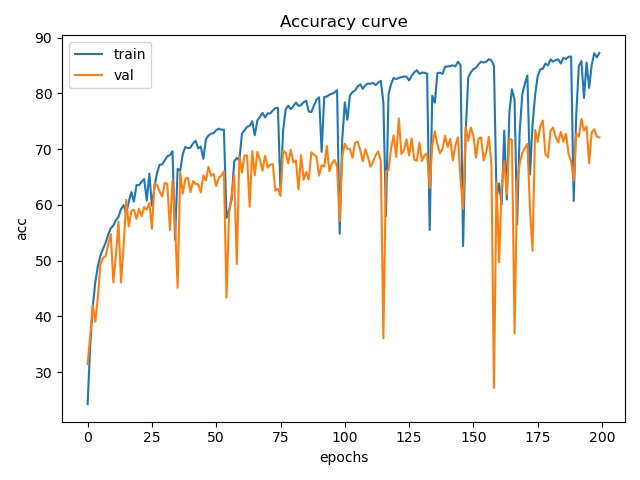
Pseudo label：



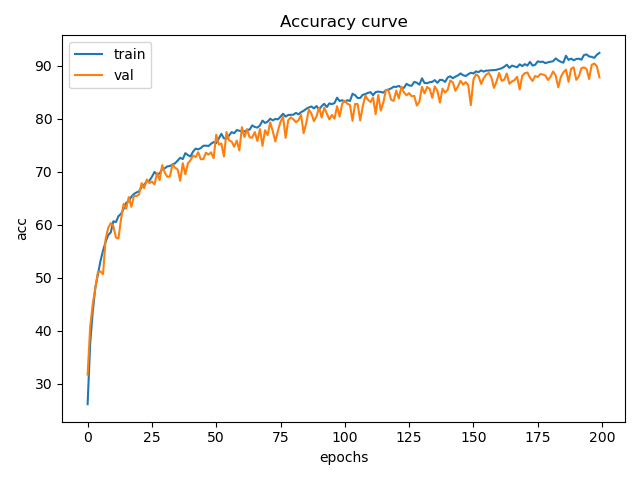
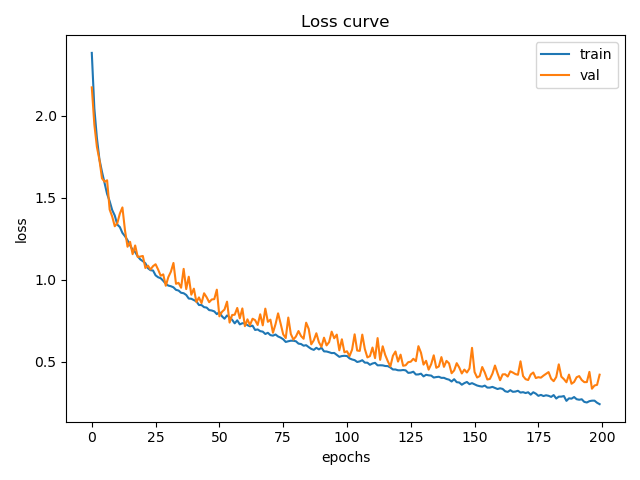


MixMatch：

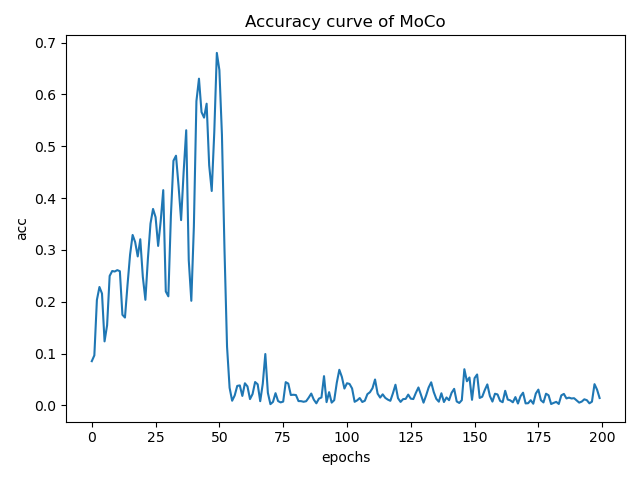
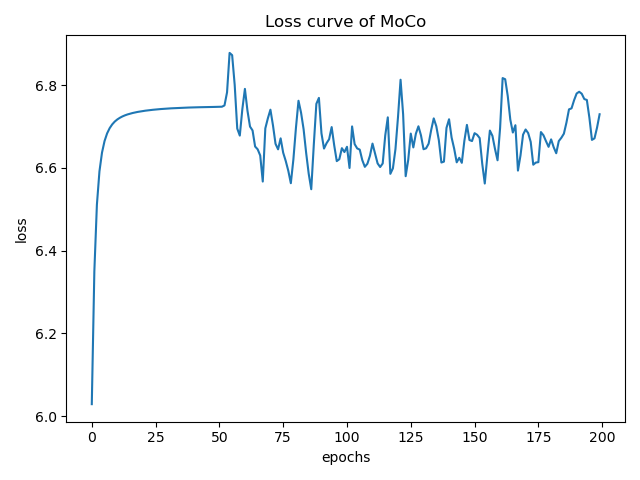




Supervised：



原版MoCo：



改版MoCo：

