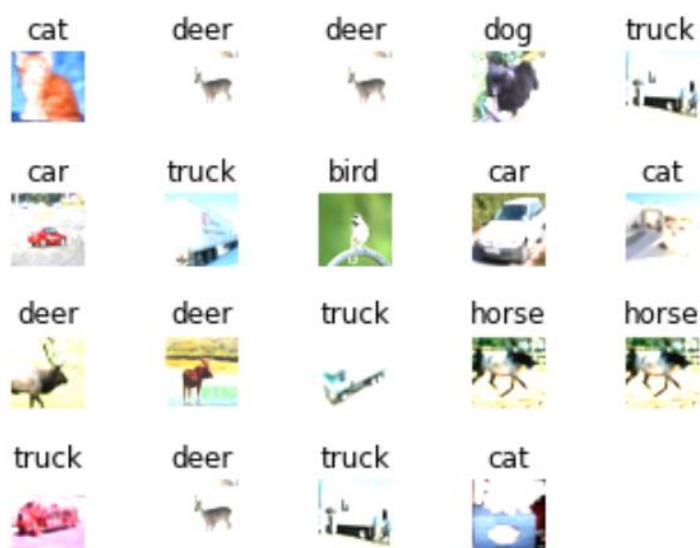


Ex9 record and report

王敏行 id: 2018012386 wangmx18@mails.tsinghua.edu.cn

本次实验利基与 Python 和 PyTorch，用卷积神经网络（CNN）对 [CIFAR-10](#) 数据集进行物体识别的分类。本次实验总共尝试了从简单到复杂的两个网络。CIFAR-10 数据集是 10 类物品的图片，图片尺寸为 $32 \times 32 \times 3$ 。本次的任务就是根据 $32 \times 32 \times 3$ 维的数据输入，对图片进行分类，如下图所示。



分类器

本文一共构建了两个 CNN 网络，分别记作 CNN1、CNN2。

CNN1 是本实验中最简单的网络，其结构为‘卷积-池化-卷积-线性全连接-线性全连接’，层与层之间利用 sigmoid 函数进行非线性连接。CNN2 参考自[简书](#)，层数更多且复杂。

训练策略

本实验是一个多分类实验，用交叉熵 `torch.nn.CrossEntropy()` 作为代价函数对分类的结果进行量化。训练方面，采用随机梯度下降方法 `torch.optim.SGD()` 进行优化权重，选取学习率下降、梯度动量参数。具体的参数设置见结果。考虑到设备本身内存有限，采取 minibatch 方法进行训练。

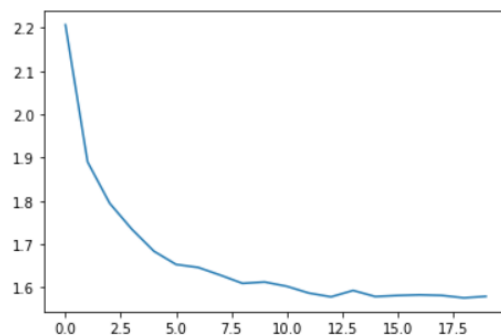
实验结果

CNN1 和 CNN2 训练了 20 个 epoch，前者用时约 3min，后者用时约是 7 min。测试集上的准确率已经呈现在途中。以下结果都是可以重复的。

对 CNN1 的进行训练时共采取了两种超参数设置：

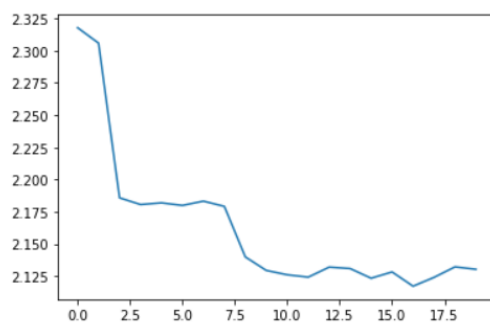
1. 学习率：1，梯度动量：0.9，不采取学习率下降（optimizer1）。

Accuracy of the network is: 42.2500 %



2. 学习率：1，梯度动量：0.9，学习率下降：0.09（越大，下降越快）（optimizer3）。

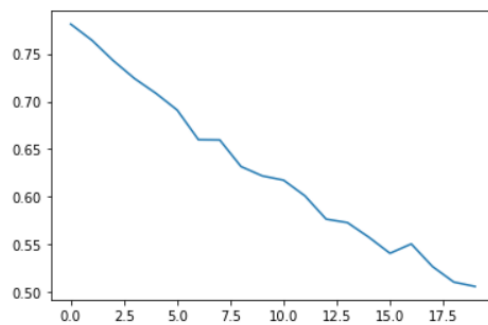
Accuracy of the network is: 19.7400 %



对 CNN2 也采用两种超参数设置：

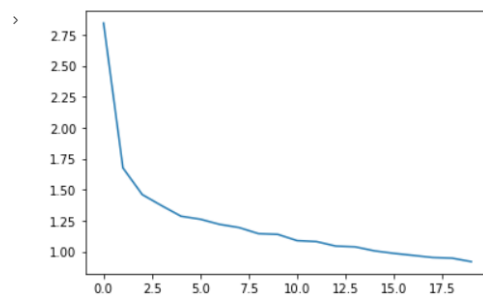
1. 学习率：0.01，梯度动量：0.9，不采取学习率下降（optimizer2）。

Accuracy of the network is: 71.7900 %



2. Adam 方法优化，学习率：0.01，不采取学习率下降（optimizer4）。

Accuracy of the network is: 64.3700 %



讨论

首先，对比两个模型，后者的准确率显然更好。但是大模型也就意味着对于算力的要求更高。本次实验中，CNN2 训练时间是 CNN1 的约 2 倍。

在学习率较大的情况下，有可能会发生模型参数在最优点附近振荡，从而导致最终 loss 会有波动或者没能收敛到更低的值。显然，这样的问题可以通过减小学习率解决，但会导致收敛需要 epoch 大量增加。更为恰当的解决方法可以是引入梯度的动量以避免梯度的突然变化，或者加入学习率衰减，使得 epoch 越大学习率越小。

Optimizer1 和 3 的比较说明引入学习率衰减也会导致学习需要的 epoch 数目增加，由于设备性能受限，没能运行至 optimizer3 训练的模型完全收敛。

Optimizer2 和 4 的比较说明补采用梯度动量训练的话，通常会引起模型收敛的极限较大，引入动量可以有效减少模型的 loss 的极限。

感谢农宇涵（数据加载和模型测试部分收到了来自他的帮助）、bing.com、简书（CNN2 模型修改自[这个博客](#)）、知乎和 CSDN，虽然找到的东西有近一半是没啥用的，但是把所有的解决方案都试一遍还是可以解决问题的。