## 使用VIT完成CAFIR10分类

## 1 概述

任务：使用VIT完成CAFIR10分类。

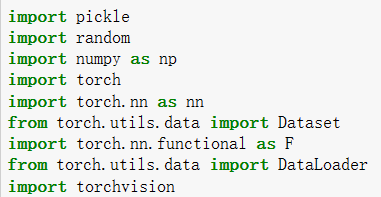
数据集：CAFIR10。CAFIR10数据集由50000张训练集图片与10000张测试集图片构成，图片为3\*32\*32 RGB图像。

解决方案：使用torchvision加载CIFAR10数据集，构建VIT网络，损失函数使用交叉熵损失，优化器使用Adam。

2 解决方案

2.1 导入相关库：

实验使用pytorch框架，torchvision用于下载与加载数据集。pickle为了打开CIFAR10的原数据集，会对数据做一些特殊处理



2.2 获取数据集

首先创建对图片进行预处理的transform，将图片转换为tensor并进行归一化处理，这里按照均值为0.5，方差为0.5进行处理。

这里创建了个CIFAR\_Dataset类用于读取CIFAR10数据并做处理，最终返回训练集与测试集。

最后创建DataLoader对象，用于训练与测试，训练集对象与测试集对象分别传入训练集与测试集，batch\_size代指批处理大小，shuffle=True表示对数据随机打乱。

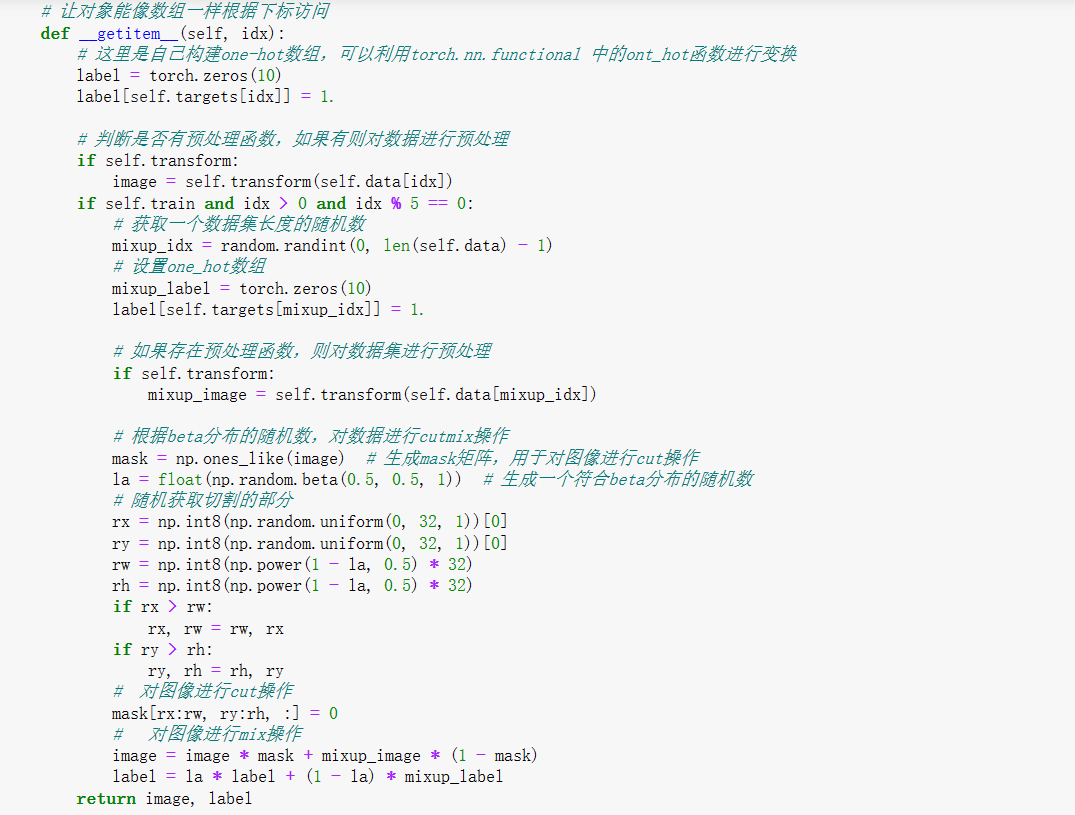


2.3 CIFAR\_Dataset类

这里将原CIFAR10数据读了出来，并对训练集做了处理。

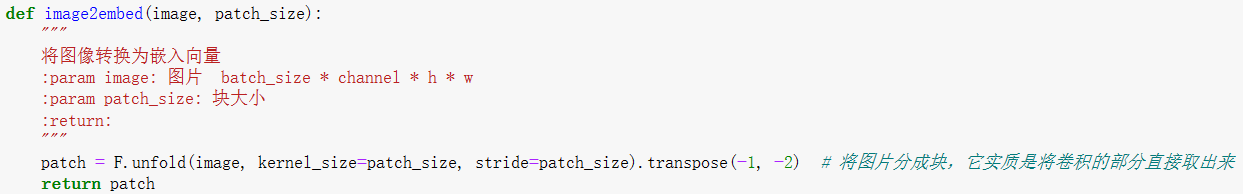
这里的处理方式是随机抽取一部分数据，将它与其它数据进行cutmix操作，用于对数据的增强。





2.4 取图片中的patch

这里将patch取出来用于后面的Embedding



2.5 Embedding 层

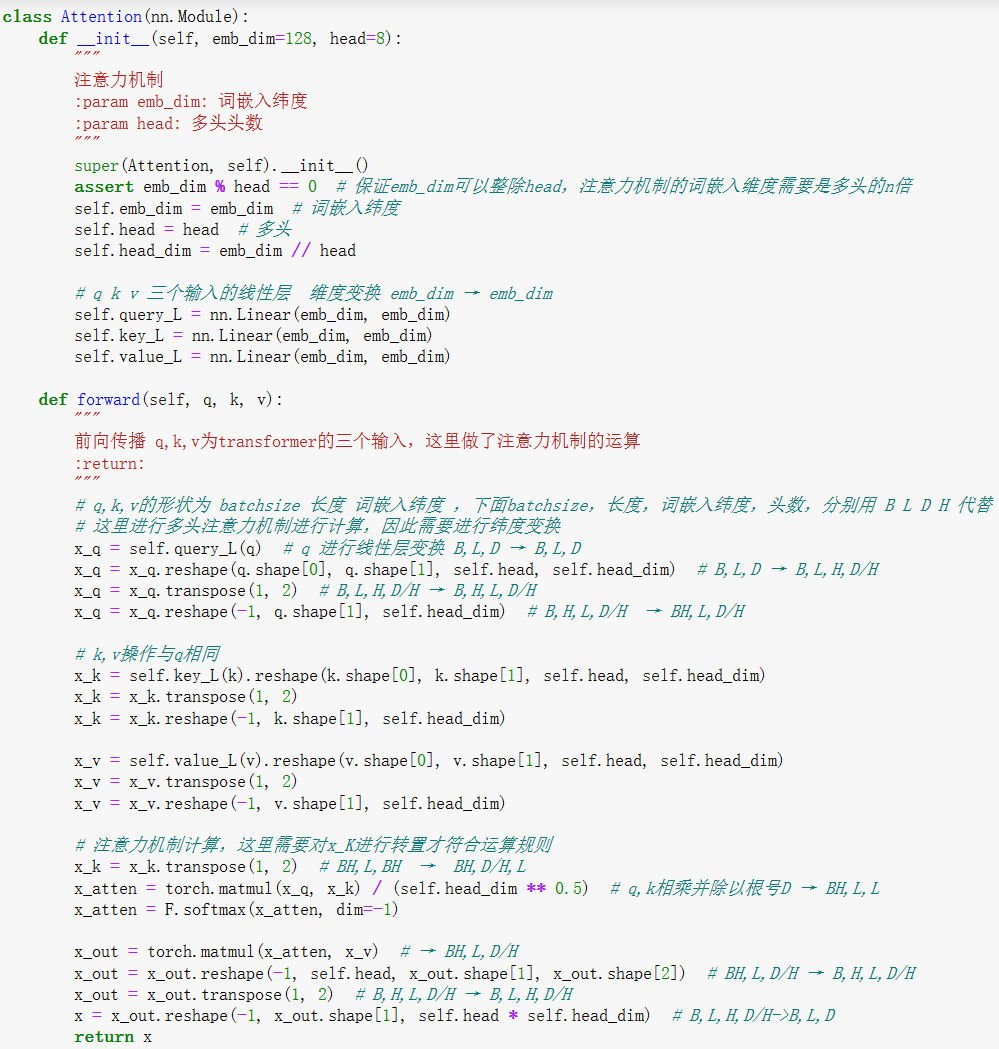
将patch块获取到之后，对数据进行Embedding与位置编码。在这里加入了一个CLS\_token，

它的作用是获取全局的注意力，它可以插入到一条数据的任何位置，在输出时，取这个位置 的结果用表示预测结果。同时，这里也嵌入了位置编码。



2.6 注意力类

这里实现了注意力机制的算法，因为没有需要掩码的部分，所以这里去除了掩码的设计。



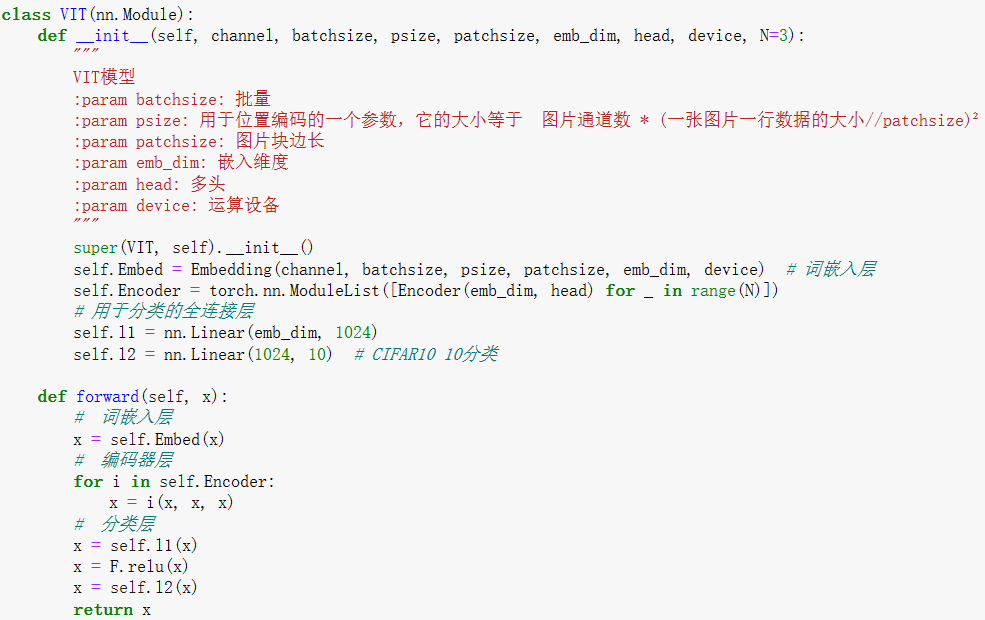
2.7 Encoder类

这里实现了编码器类。



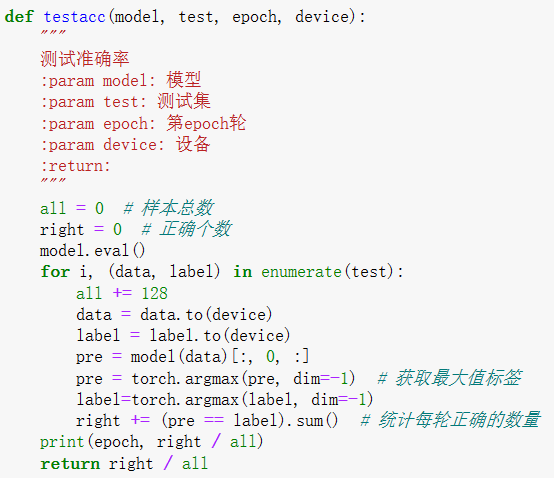
2.8 VIT模型

这里实现VIT模型。数据的处理流程如下，首先进行位置编码与词嵌入，然后在N层Encoder中算注意力与传播，最后进行10分类操作。



2.9 定义准确率函数

与手写数字识别中定义的函数基本相似，不过这里将cls\_token所在位置的输出作为模型的预测。



2.10 定义训练函数

与手写数字识别中定义的函数基本相似，模型换成VIT，输出为cls\_token所在位置的输出。



3 总结

训练参数如下：优化器：Adam

损失函数：交叉熵

学习率：0.0003

batch\_size:128

epochs:30

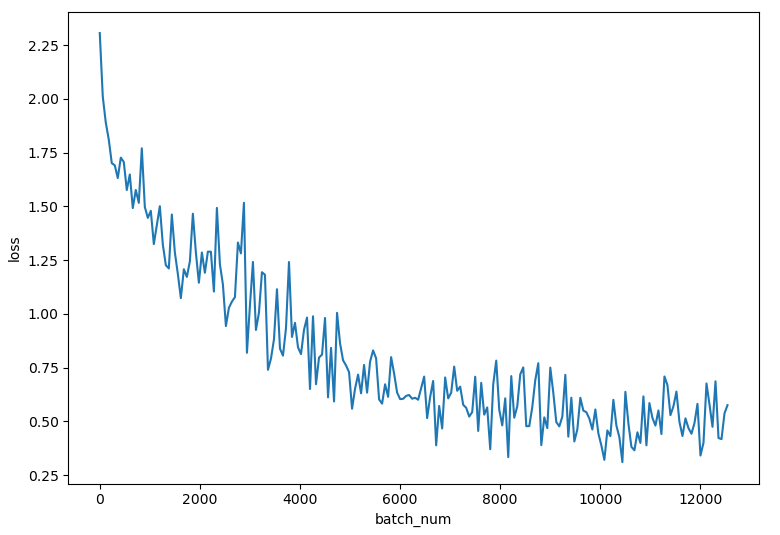
训练设备:GPU

Encoder层数：6

patch大小：4

模型最终的准确率会收敛到61%左右。不过可能我打VIT模型较小，最好的结果目前也才只达到了63%。但是我使用Resnet-20时，准去率能轻松突破70%。可能是模型参数没有调整好，或者Encoder层数太少。

Loss值下降曲线



准确率上升曲线：

