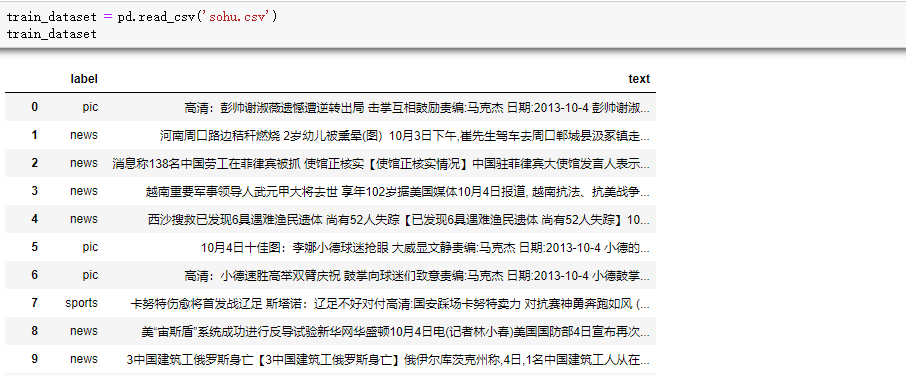
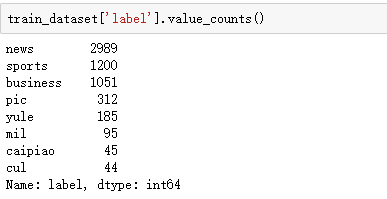
1. 基于RNN实现文本分类任务

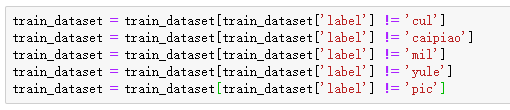
1.读取数据



2.把数据集里的label列按出现顺序排序



3.由于’pic’,’yule’,’mil’,’caipiao’,’cul’出现较少，从数据集中删去



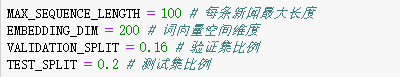
4.将news，sports，business三列映射为数字



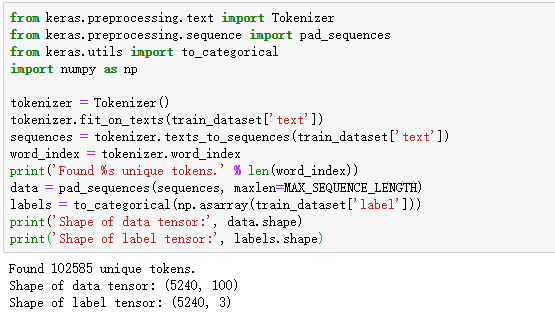
5.利用jieba库进行分词处理



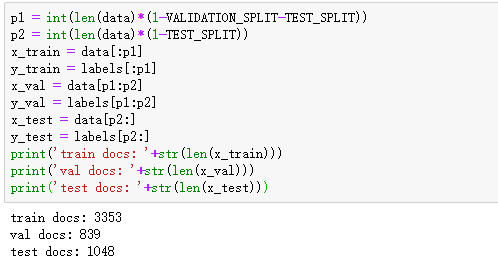
6.设置每条新闻的长度，词向量的空间维度，验证集及测试集比例



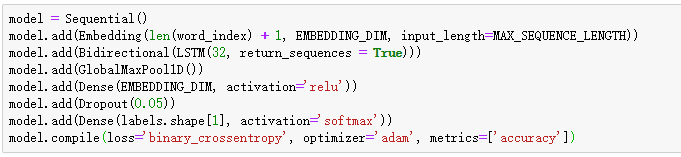
7. 第一步先把训练与测试数据放在一起提取特征，使用 keras 的 Tokenizer 来实现，将新闻文档处理成单词索引序列，单词与序号之间的对应关系靠单词的索引表 word\_index 来记录，这里从所有新闻中提取到 102585 个单词，然后将长度不足 100 的新闻用 0 填充（在前端填充），用 keras 的 pad\_sequences 实现；最后将标签处理成 one-hot 向量，比如 6 变成了 [0,0,0,0,0,0,1,0,0,0,0,0,0]，用 keras 的 to\_categorical 实现



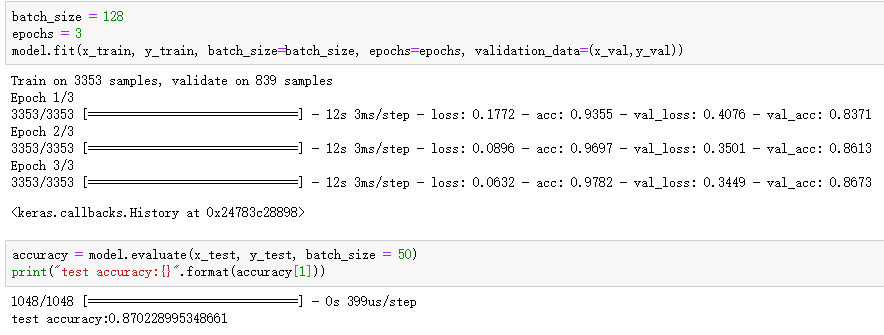
8.分割出训练集，测试集和验证集



9.构建模型



10.设置batch\_size,epochs,开始训练模型，并求出准确率为87%



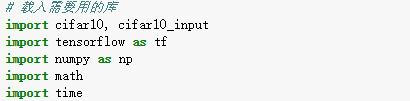
1. 基于CIFAR-10数据集使用CNN完成图像分类任务。

CIFAR-10 是一个包含60000张图片的数据集。其中每张照片为32\*32的彩色照片，每个像素点包括RGB三个数值，数值范围 0 ~ 255。所有照片分属10个不同的类别，分别是 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck'。其中五万张图片被划分为训练集，剩下的一万张图片属于测试集。

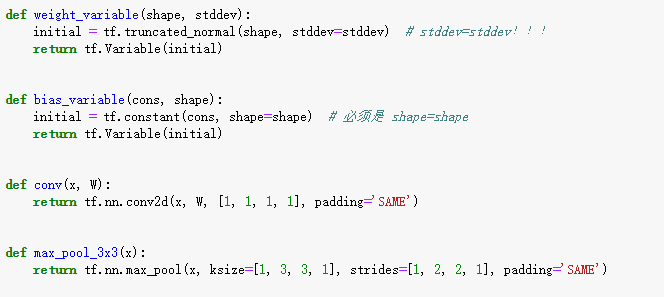
1.git clone https://github.com/tensorflow/models.git

下载cifar10\_input.py （读取本地CIFAR-10的二进制文件格式的内容）和 cifar10.py（建立CIFAR-10的模型）这两个文件。

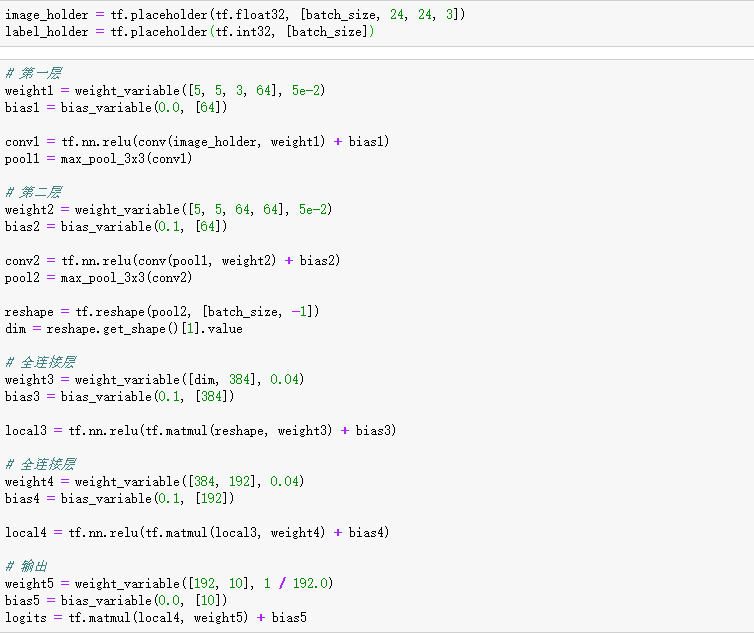
2.导入所需要的库



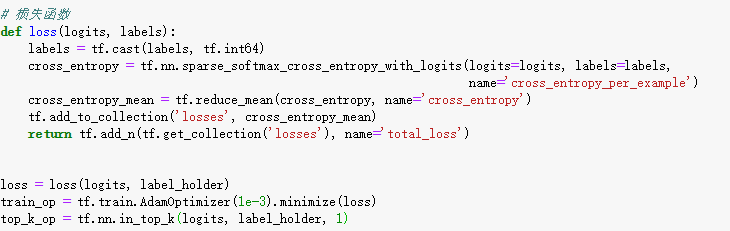
3. 参数设置



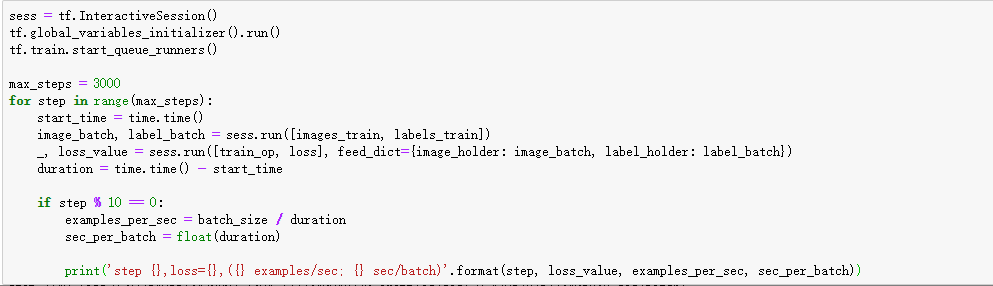
4. **模型实现：首先创建输入数据的 placeholder，在设定时需要注意，因为 batch\_size 在之后定义的网络结构时被用到了，所以数据尺寸中的第一个值（样本条数）需要被预先设定，而不是像之前那样设定为 None。而数据尺寸中的图片尺寸为剪裁后的 24x24，因为图片是彩色 RGB 图片，所以通道数为 3。**



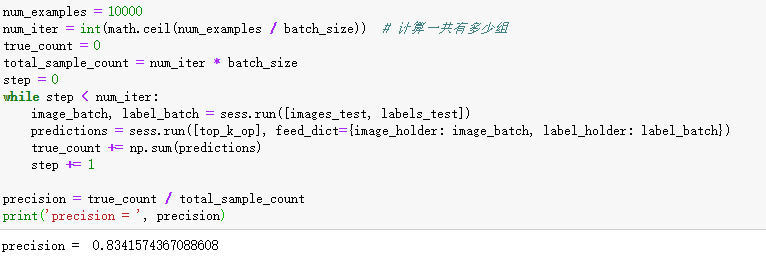
**5.** **损失函数：依然使用 cross\_entropy 作为损失函数，不同的是在这里将 cross\_entropy 的计算和 softmax 的计算混合在了一起，使用 tf.nn.sparse\_softmax\_cross\_entropy\_with\_logits。然后将 cross\_entropy 的均值添加到 total\_loss 的 collection 中（后面会加入 L2 正则，所以计算 total\_loss）。最后，使用 tf.add\_n 将 collection 中的 loss 全部求和，得到最终的 loss。优化器使用 Adam Optimizer。使用 tf.nn.in\_top\_k 函数输出结果中 top k 的准确率，默认使用 top 1，也就是输出分数最高的那一类的准确率。**



6. **训练：**如下代码中的第三行启动前面提到的图片数据增强的线程队列，这里一共使用了 16 个线程来进行加速。注意，如果这里不启动线程，那么后续的 inference 及训练的操作都是无法开启的。



7. 评测准确率：在每一个 step 中使用 session 的 run 方法获取 images\_test、labels\_test 的 batch，再执行 top\_k\_op 计算模型在这个 batch 的 top 1 上预测正确的样本数。

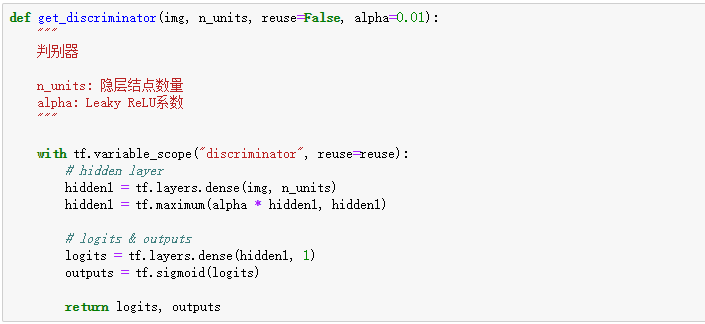
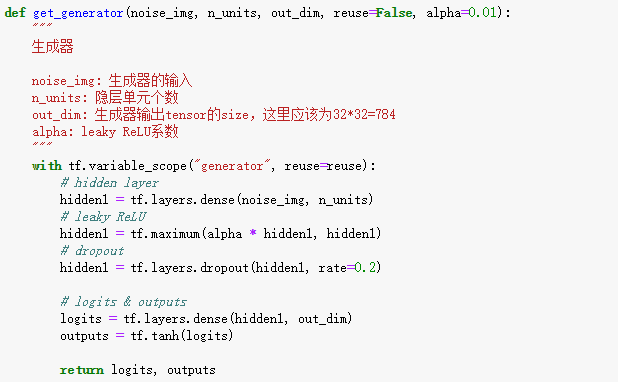
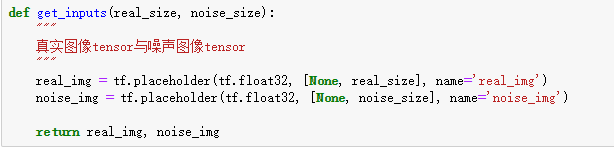


1. **基于MNIST数据集使用GAN实现手写图像生成的任务**

1.导入数据



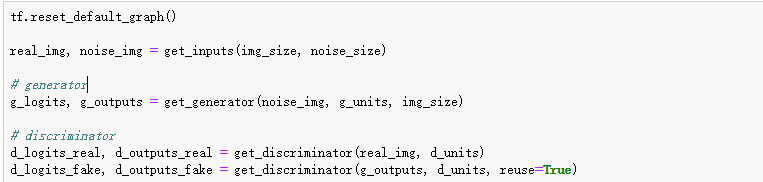
2.构建模型：inputs,generator,discriminator



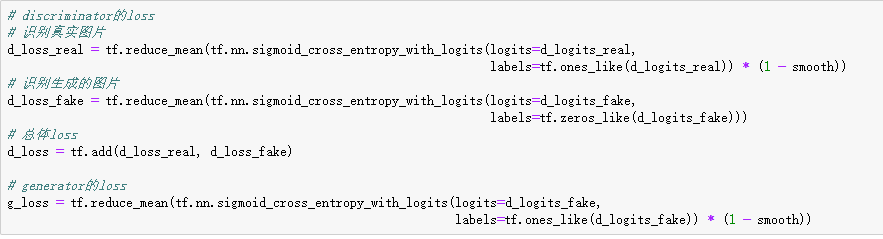
3.定义参数



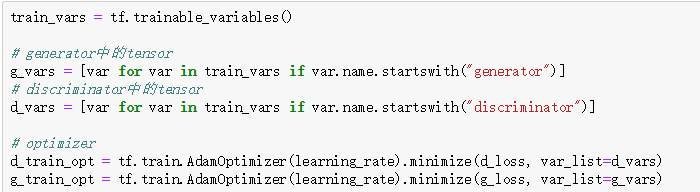
5.构建网络



6.Loss：generator和discriminator，因此需要分别计算loss。discriminator的目的在于对于给定的真图片，识别为真（1），对于generator生成的图片，识别为假（0），因此它的loss包含了真实图片的loss和生成器图片的loss两部分。generator的目的在于让discriminator识别不出它的图片是假的，如果用1代表真，0代表假，那么generator生成的图片经过discriminator后要输出为1，因为generator想要骗过discriminator。

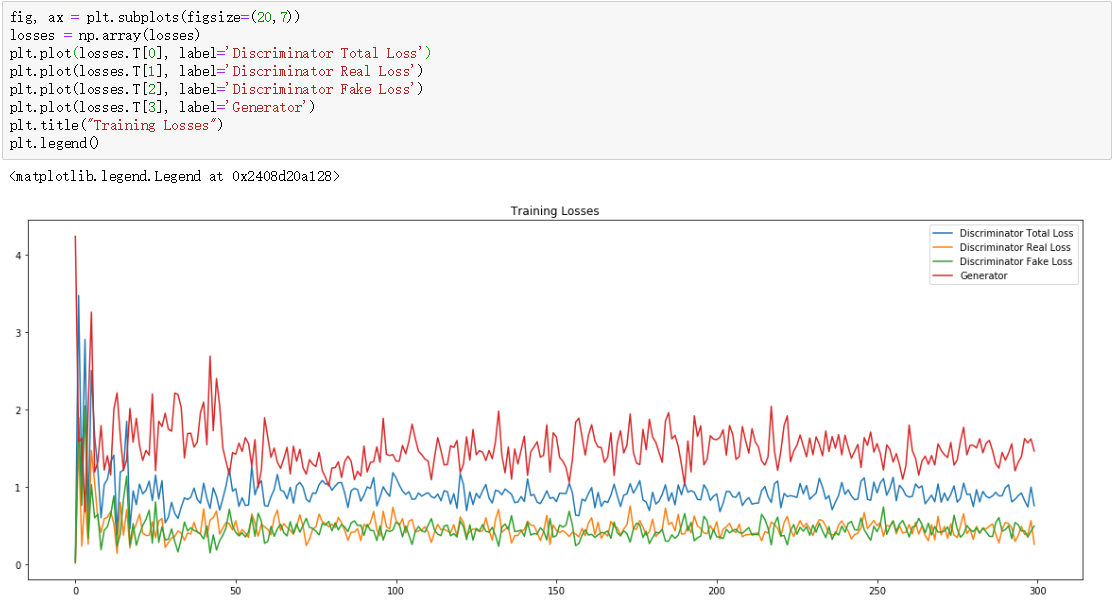


7.optimizer: 由于我们在GAN里面一共训练了两个网络，所以需要分别定义优化函数。

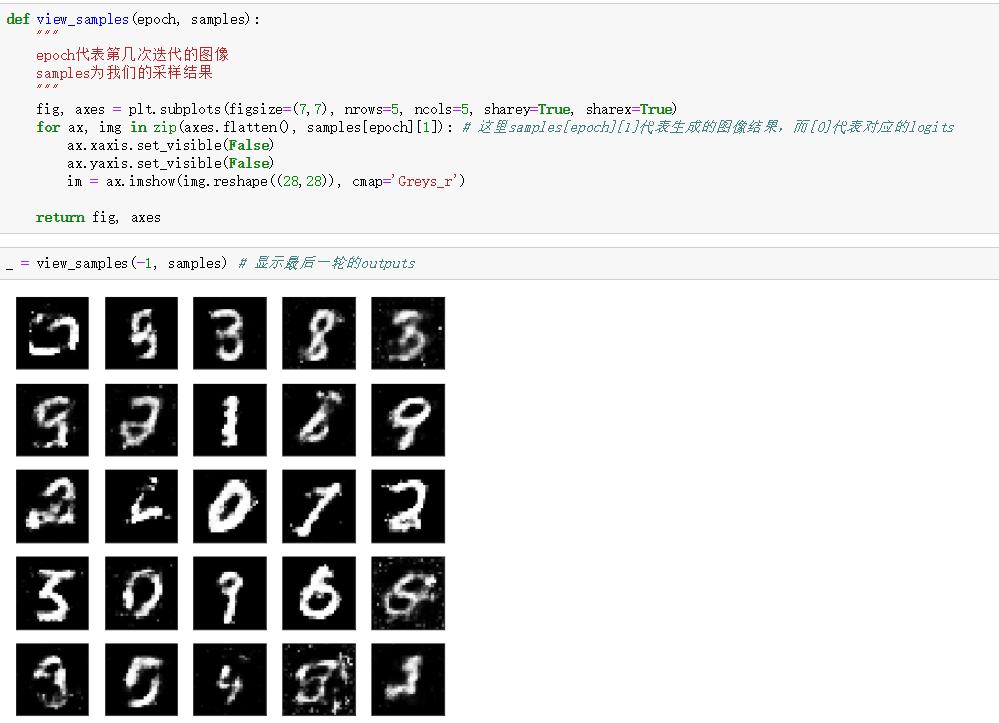


8.训练

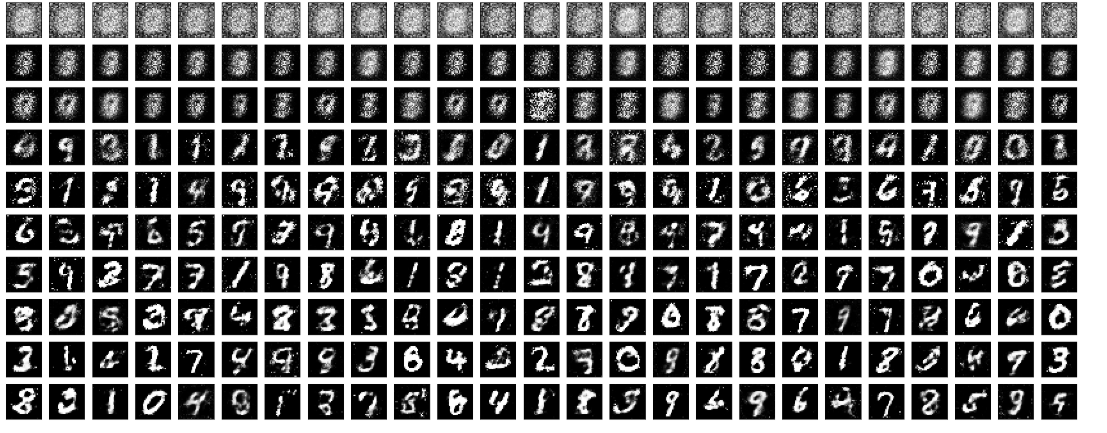
9.绘制loss曲线



10.显示图像



11.显示整个生成过程的图片



12.生成新的图片

