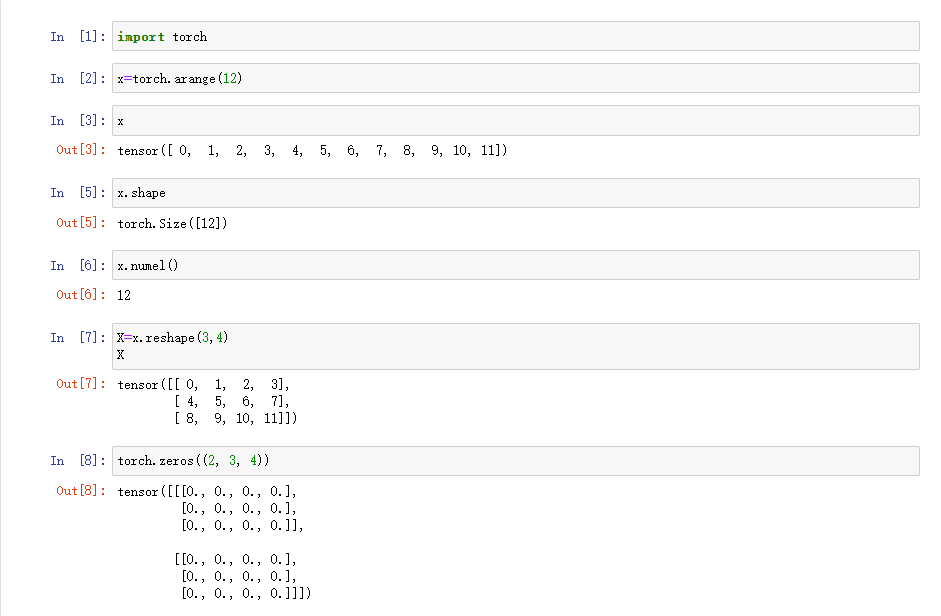
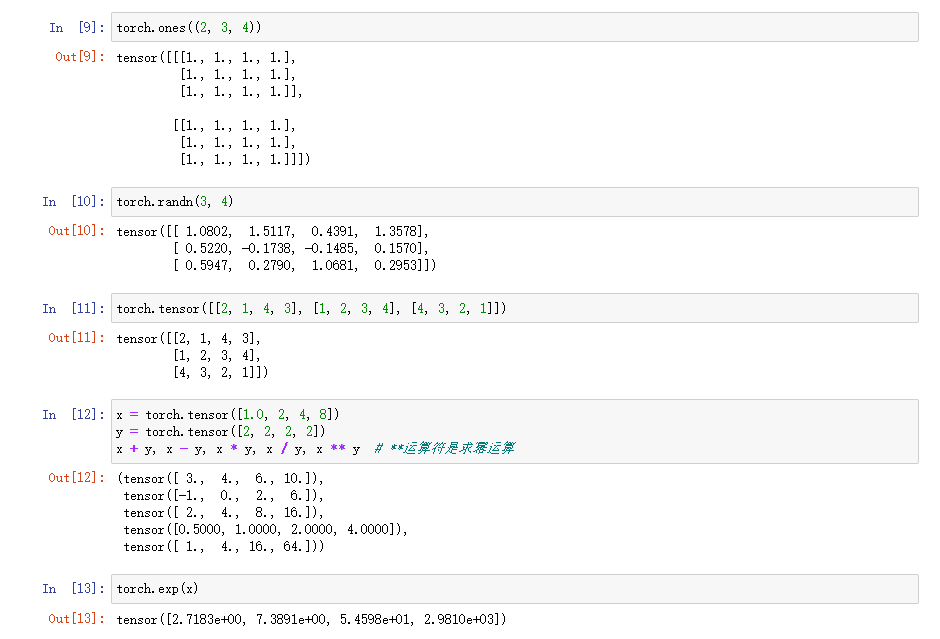
**深度学习第一次作业报告**

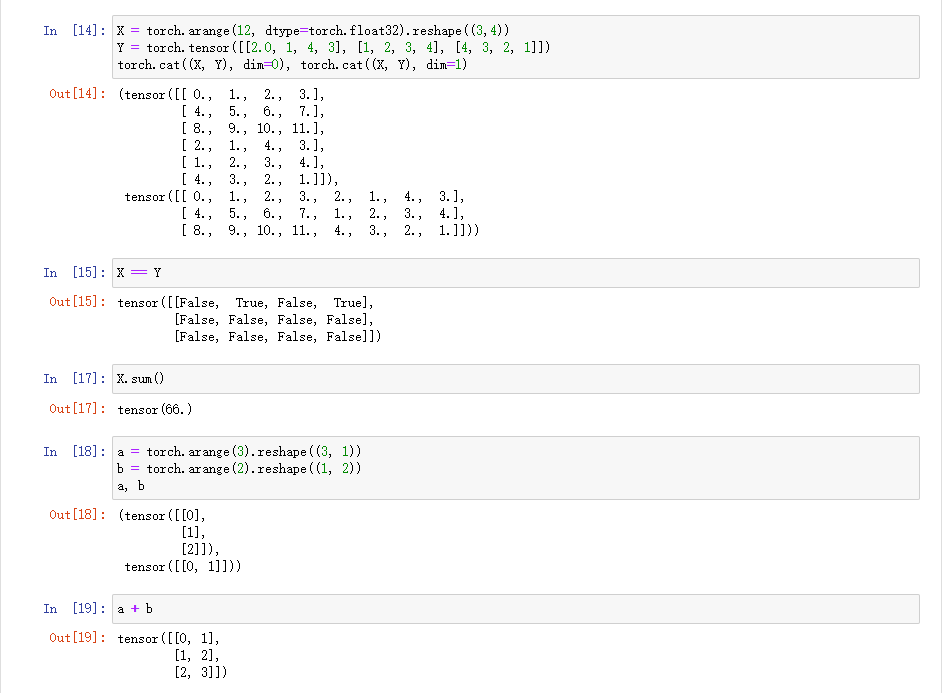
**2020311448-胡雅娴**

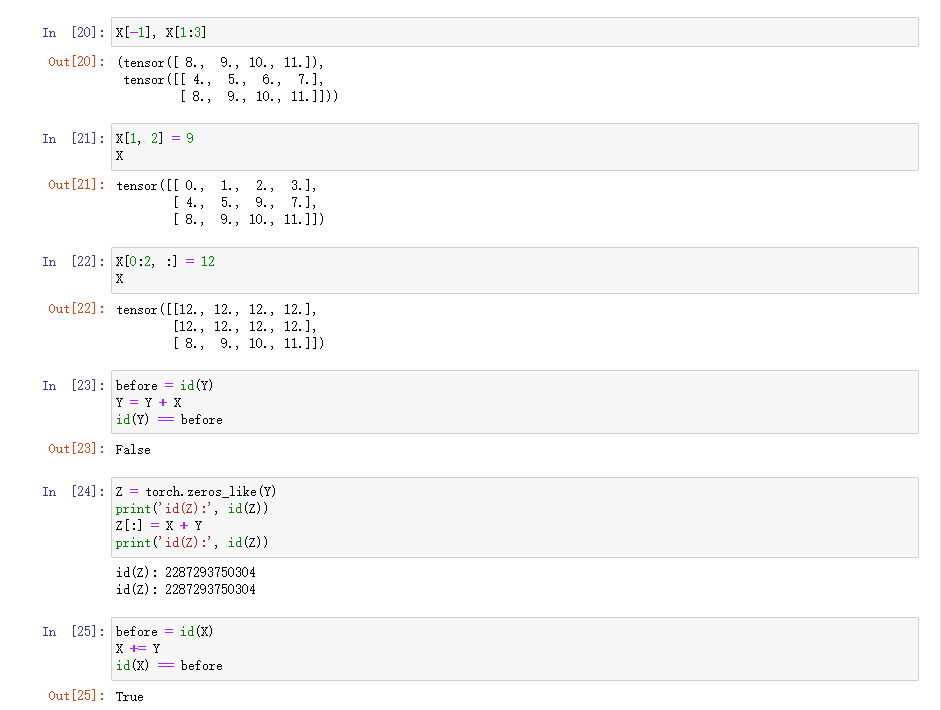
**3.**

（1）运行本节中的代码，结果如下：





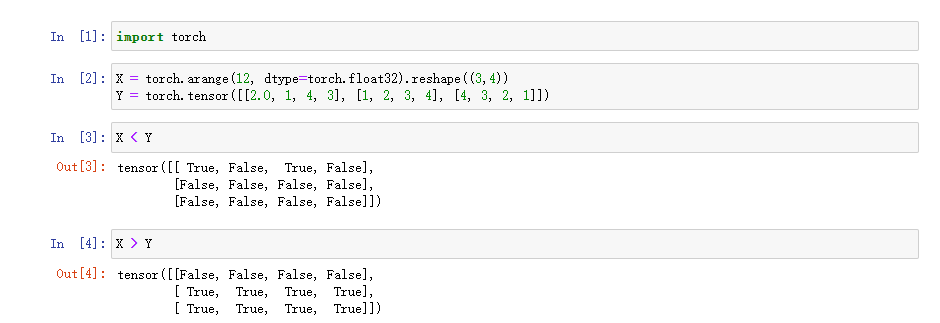






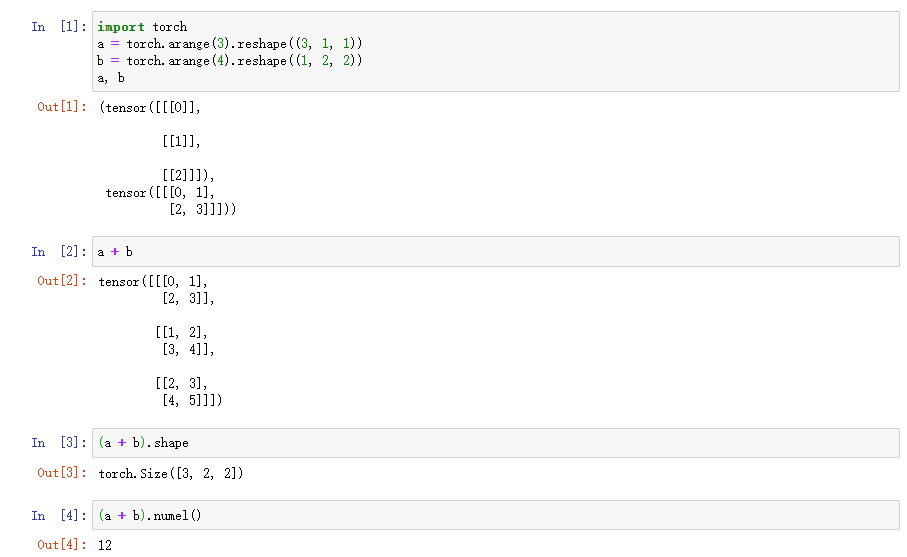
以上代码见文件 深度学习2.1.1。

将本节中的条件语句X == Y更改为X < Y或X>Y，得到这样的张量：



以上代码见文件 深度学习2.1.1.2。

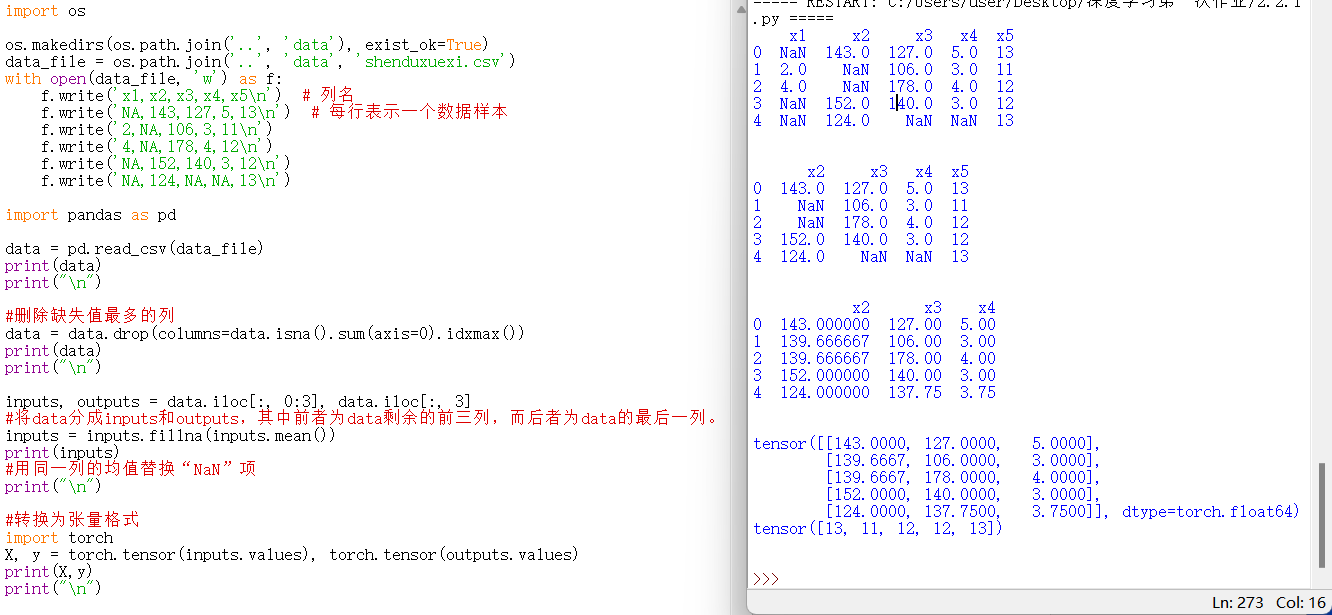
（2）用其他形状（例如三维张量）替换广播机制中按元素操作的两个张量，得到这样的结果：



以上代码见文件 深度学习2.1.2。

**4.**

（1）（2）代码及运行结果如下：

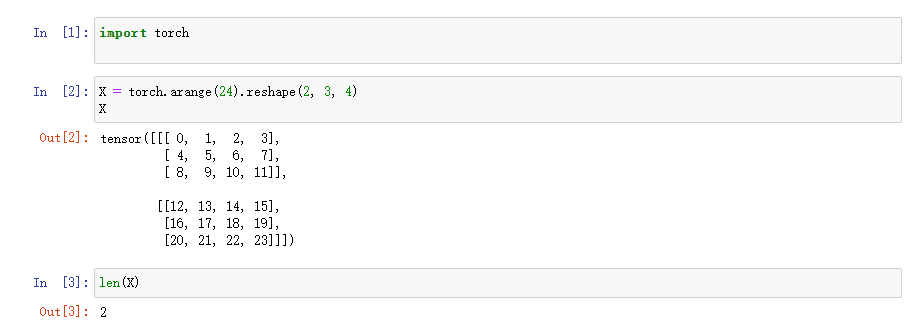


代码见文件 深度学习2.2.1。

**5.**

（4）本节中定义了形状（2，3，4）的张量X。len(X)的输出结果是什么？

输出结果是2。代码运行如下：

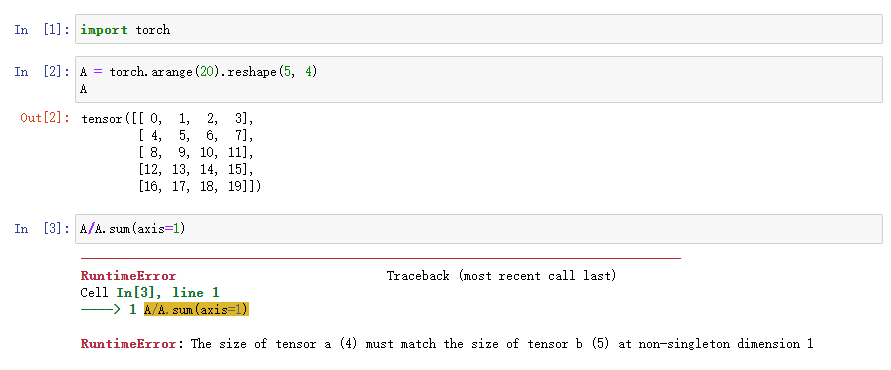


（5）对于任意形状的张量X,len(X)是否总是对应于X特定轴的长度?这个轴是什么?

是的，对应轴axis=0.

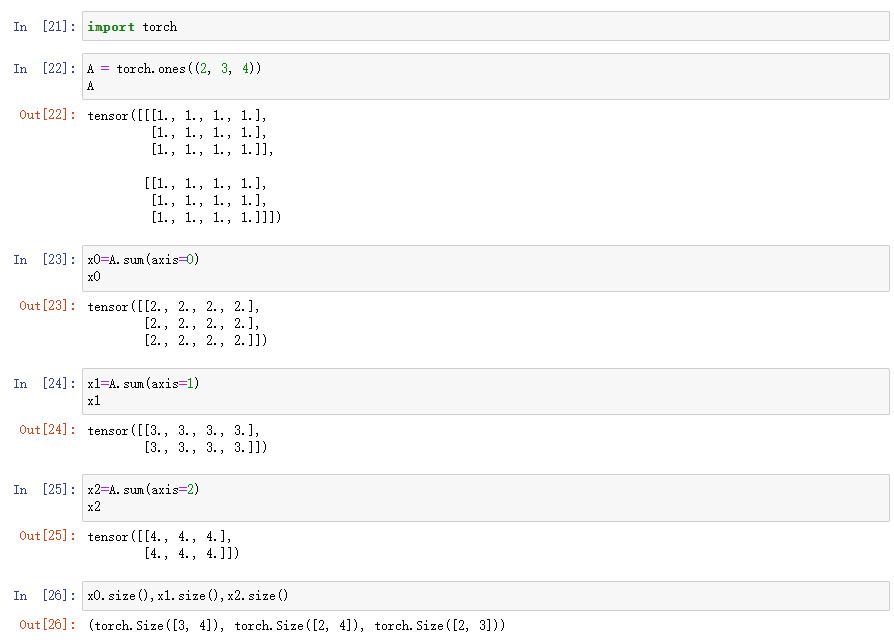
（6）运行A/A.sum(axis=1)，看看会发生什么。请分析一下原因？

运行该代码会报错，这是由于广播机制中，它们的形状不匹配。代码运行如下：



（7）考虑一个具有形状（2，3，4）的张量，在轴0、1、2上的求和输出是什么形状?

形状分别为[3, 4], [2, 4], [2, 3].代码运行如下：



（8）为linalg.norm函数提供3个或更多轴的张量，并观察其输出。对于任意形状的张量这个函数计算得到什么?

这个函数计算得到二范数。代码运行如下：



**6.**

（2）函数 的梯度是 .

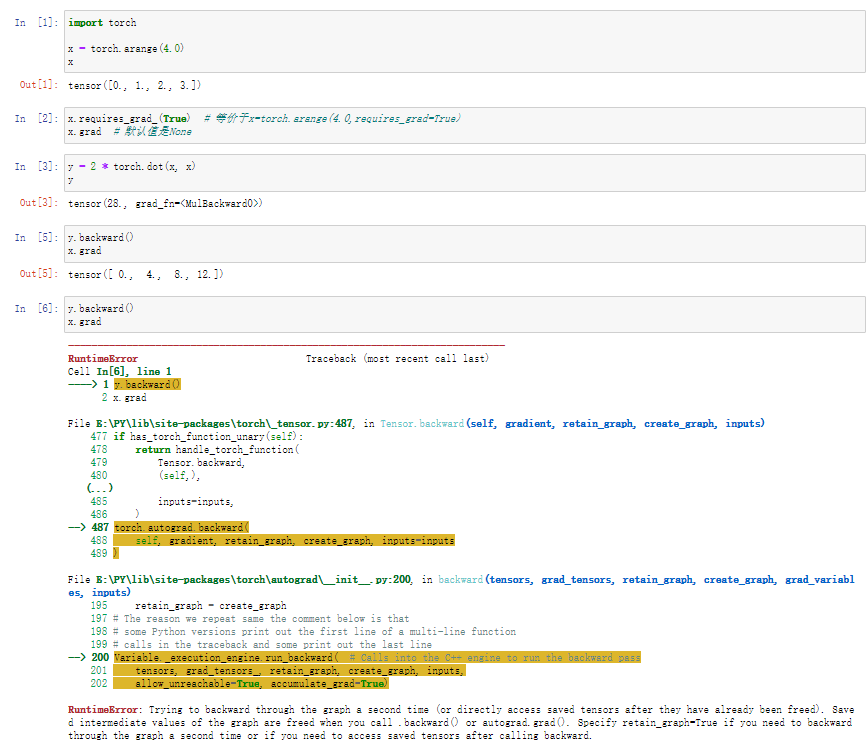
（3）函数 的梯度是 .

（4）链式法则为：

**7.**

（2）在运行反向传播函数之后，立即再次运行它，看看会发生什么。

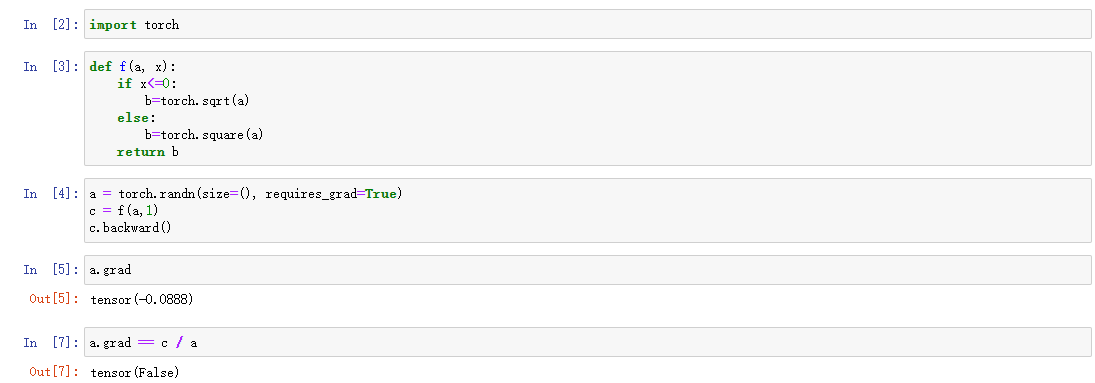
会报错（Trying to backward through the graph a second time (or directly access saved tensors after they have already been freed). Saved intermediate values of the graph are freed when you call .backward() or autograd.grad(). Specify retain\_graph=True if you need to backward through the graph a second time or if you need to access saved tensors after calling backward.），因为进行反向传播后，中间变量被释放，无法进行第二次反向传播，代码运行如下：



以上代码见文件 深度学习2.5.2.

（4）重新设计一个求控制流梯度的例子，运行并分析结果。

如图所示，该函数在x<=0时求开方，否则求平方。



以上代码见文件 深度学习2.5.4。

**8.**

导入工具包和数据。

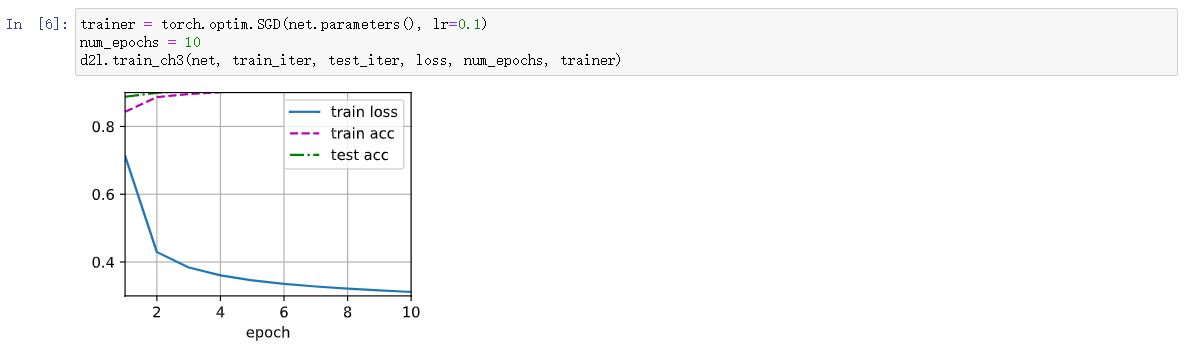


为了使我们在读取训练集和测试集时更容易，我们通过内置数据迭代器随机打乱所有样本，从而无偏见地读取小批量，将其设置为2的幂次，可以发挥更好的性能，因此选取batch\_size=256。



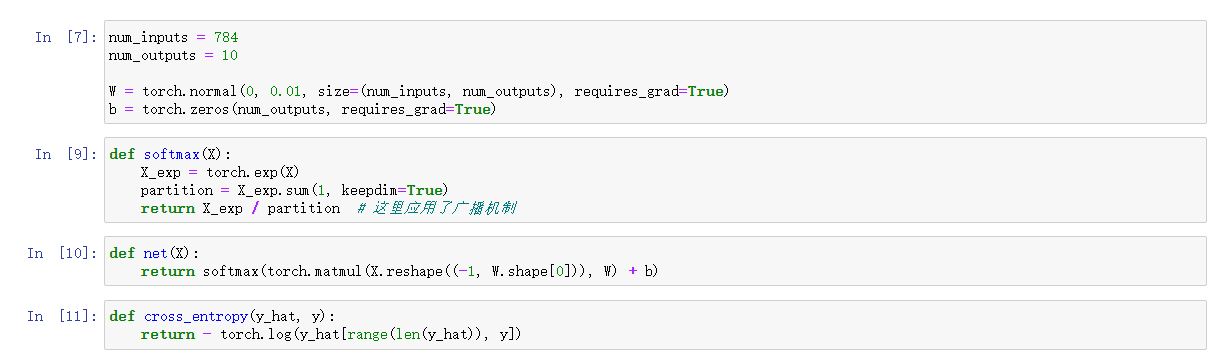
原始数据集中的每个样本都是 28×28 的图像。 展平每个图像，把它们看作长度为784的向量。因为我们的数据集有10个类别，所以网络输出维度为10。

用简洁算法进行一次实验，在交叉熵损失函数中传递未规范化的预测，并同时计算softmax及其对数，使用学习率为0.1的小批量随机梯度下降作为优化算法，可以看到，该算法的结果收敛到一个较高的精度。



由于简洁算法不易于呈现其他特征，用简单算法进行一次实验。

展平每个图像，把它们看作长度为784的向量。



定义softmax操作，将每个元素变成一个非负数，每行总和为1。定义回归模型。

引入的交叉熵损失函数。

下面计算分类精度。

将y\_hat的数据类型转换为与y的数据类型一致，将预测类别与真实y元素进行比较。

定义一个实用程序类Accumulator，对多个变量进行累加，当我们遍历数据集时，正确预测的数量和预测的总数量随着时间的推移而累加。



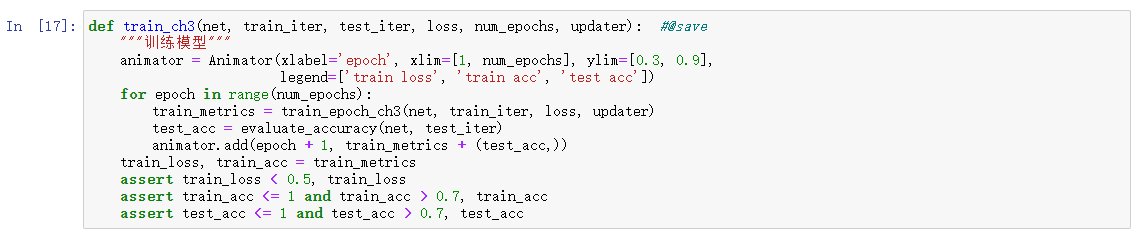
下面进行训练。定义一个函数来训练一个迭代周期。



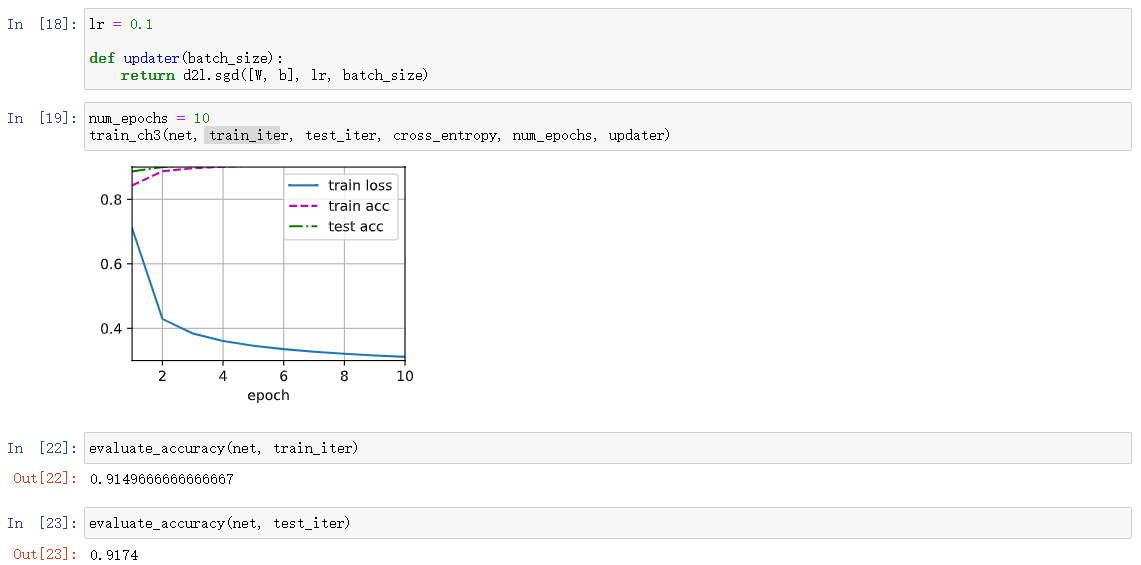
定义一个在动画中绘制数据的实用程序类，以简化代码。



定义训练函数，运行多个迭代周期，每个迭代周期结束时，Animator将可视化训练进度。



设置学习率为0.1，迭代周期为10。绘制准确率曲线，设置不同的数值，运行后比较不同参数设置下的准确率。



可以看见，迭代10轮后，训练集准确度为0.9150，预测集准确性为0.9174，预测效果好，该模型有着高鲁棒性。

以上代码见文件 softmax。