**实验一 手写数字识别 作业报告**

1. 实验目的
2. 掌握卷积神经网络基本原理；
3. 掌握Tensorflow的基本用法以及构建卷积网络的基本操作；
4. 了解Tensorflow在GPU上的使用方法。
5. 实验要求
6. 搭建Tensorflow环境；
7. 构建一个规范的卷积神经网络组织结构；
8. 在MNIST手写数字数据集上进行训练和评估。
9. 实验原理
10. TensorFlow基本用法：

使用 TensorFlow, 必须了解TensorFlow:

* 使用图(graph) 来表示计算任务。
* 在被称之为会话 (Session) 的上下文 (context) 中执行图。
* 使用 tensor 表示数据。
* 通过 变量 (Variable) 维护状态。
* 使用 feed 和 fetch 可以为任意的操作(arbitrary operation) 赋值或者从其中获取数据。

TensorFlow 是一个编程系统, 使用图来表示计算任务。图中的节点被称之为 op (operation 的缩写)。一个 op 获得 0 个或多个 Tensor, 执行计算, 产生 0 个或多个 Tensor。每个 Tensor 是一个类型化的多维数组。例如, 你可以将一小组图像集表示为一个四维浮点数数组, 这四个维度分别是 [batch, height, width, channels]。

一个 TensorFlow 图描述了计算的过程。为了进行计算, 图必须在“会话”里被启动。 “会话“将图的 op 分发到诸如 CPU 或 GPU 之类的设备上, 同时提供执行op的方法。 这些方法执行后, 将产生的tensor返回。在 Python 语言中, 返回的 tensor 是 numpy ndarray 对象; 在 C 和 C++ 语言中, 返回的 tensor 是 tensorflow::Tensor 实例。

1. 卷积神经网络：

典型的卷积神经网络由卷积层、池化层、激活函数层交替组合构成，因此可将其视为一种层次模型，形象地体现了深度学习中“深度”之所在。

* 卷积操作

卷积运算是卷积神经网络的核心操作，给定二维的图像I作为输入，二维卷积核K， 卷积运算可表示为：

 (1)

给定5×5输入矩阵、3×3卷积核，相应的卷积操作如图1所示。



图1 卷积运算

在使用TensorFlow等深度学习框架时，卷积层会有padding参数，常用的有两种选择，一个是“valid”，一个是“same”。前者是不进行填充，后者则是进行数据填充并保证输出与输入具有相同尺寸。

构建卷积或池化神经网络时，卷积步长也是一个很重要的基本参数。它控制了每个操作在特征图上的执行间隔。

* 池化操作

池化操作使用某位置相邻输出的总体统计特征作为该位置的输出，常用最大池化（max-pooling）和均值池化（average-pooling）。池化层不包含需要训练学习的参数，仅需指定池化操作的核大小、操作步长以及池化类型。池化操作示意如图2所示。



图2 池化操作

* 激活函数层

卷积操作可视为对输入数值进行线性计算发挥线性映射的作用。激活函数的引入，则增强了深度网络的非线性表达能力，从而提高了模型的学习能力。常用的激活函数有sigmoid、tanh和ReLU函数。

1. 实验所用工具及数据集
2. 工具

Anaconda、TensorFlow

（Tensorflow安装教程参考：Tensorflow官网、Tensorflow中文社区、<https://github.com/tensorflow/tensorflow>）

1. 数据集

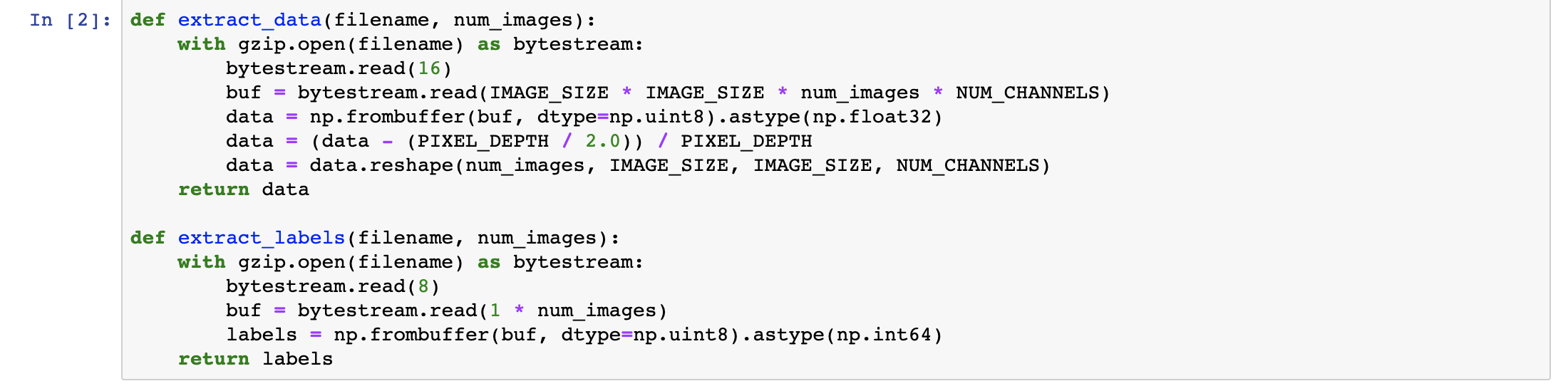
MNIST手写数字数据集

（下载地址及相关介绍：http://yann.lecun.com/exdb/mnist/）

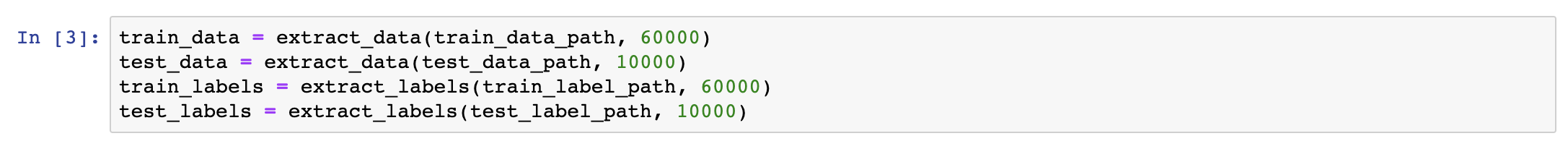
1. 实验步骤与方法

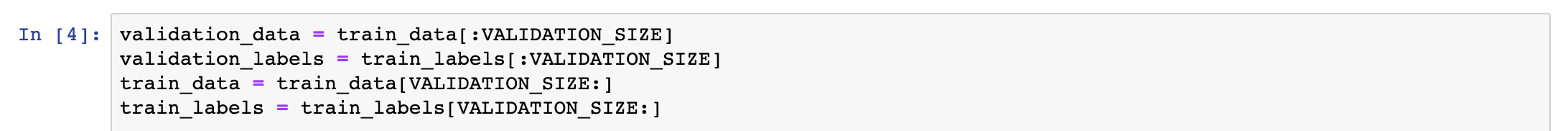
提取数据





将数据分为训练集，测试集和验证集



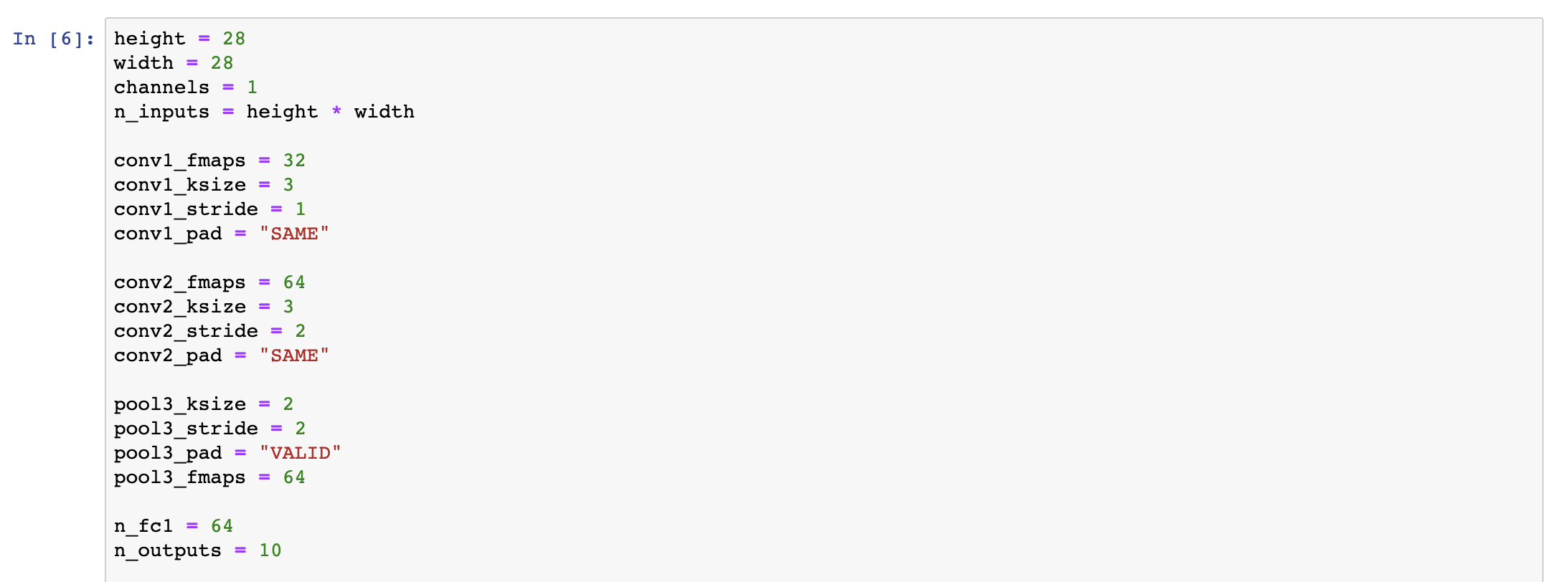


查看数据集





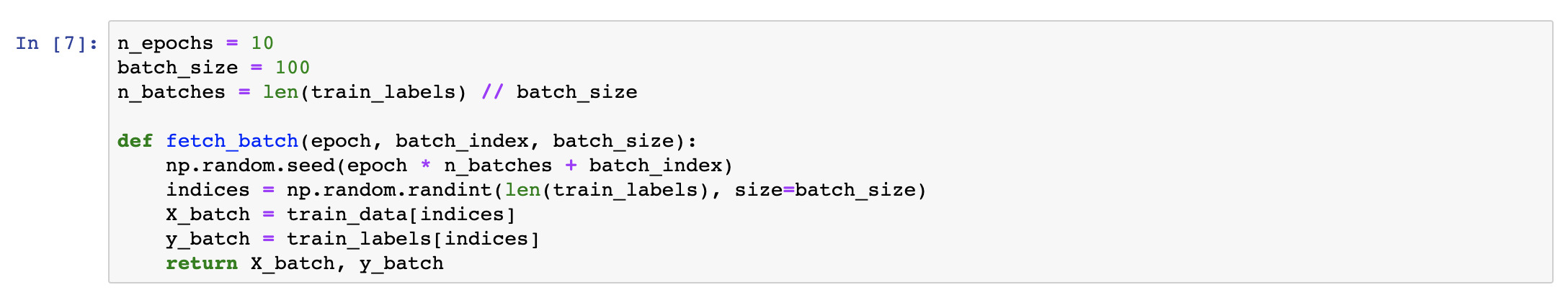
设置训练参数



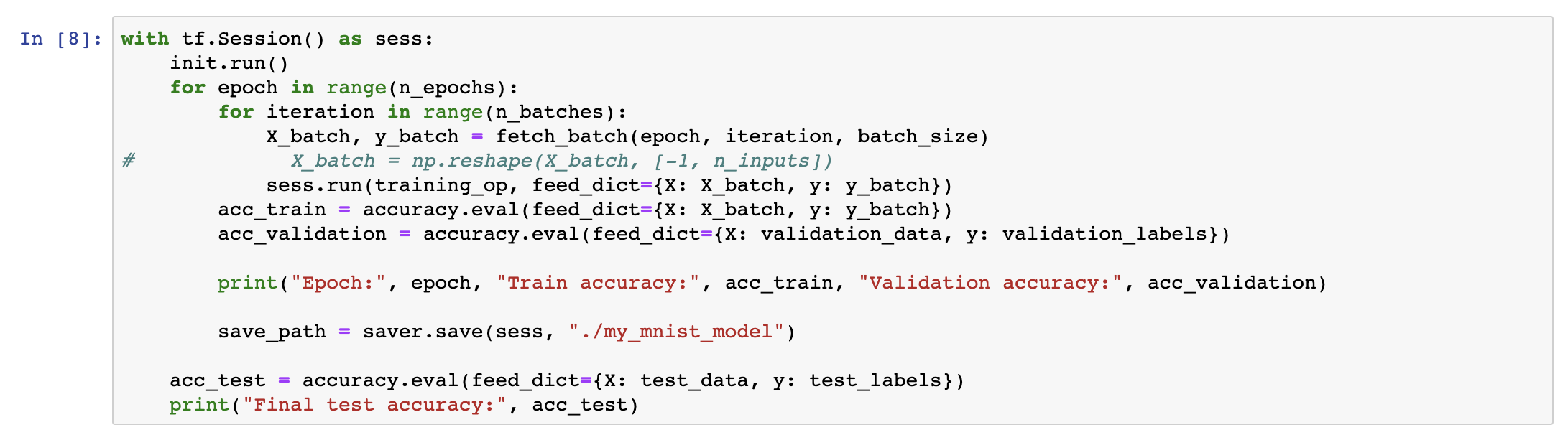
创建CNN模型



设置批量读取数据的函数，等待训练循环读取批量



训练模型，并在验证集上进行验证，最后输出测试集结果



实验最终结果展示

