北京大学
PEKING UNIVERSITY

实践课作业

2024-2025学年

任务描述：迈出第一步，做一个调包侠

北京大学 PEKING UNIVERSITY

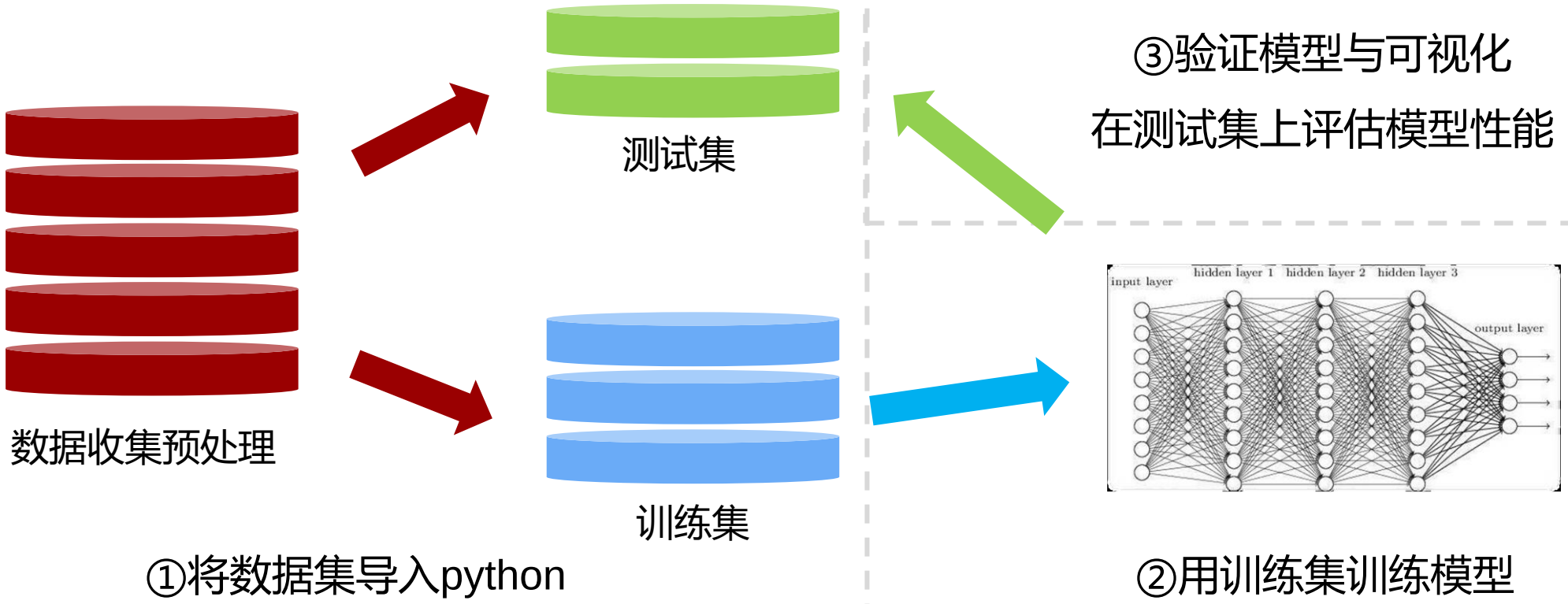
- 下列三个任务选一个，编程语言不限
  - Sklearn实现鸢尾花分类
  - Tensorflow2.0实现手写数字识别
  - Pytorch实现手写数字识别
- 提交文件：实验报告（word格式）+ 代码
- 提交方式：教学网
- 截止日期：

补充材料：机器学习基本流程（python实现）

- 以一个数据集为例（Iris数据集）
- Iris数据集是常用的分类实验数据集，由Fisher1936收集整理。Iris也称鸢尾花卉数据集，是一类多重变量分析的数据集。数据集包含150个数据样本，分为3类，每类50个数据，每个数据包含4个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Sentosa，Versicolor，Virginica）三个种类中的哪一类。其中120个作为训练集，30个作为测试集。
- 对于每个样本
  - Features：
    - Sepal.Length （花萼长度），单位是 cm;
    - Sepal.Width （花萼宽度），单位是 cm;
    - Petal.Length （花瓣长度），单位是 cm;
    - Petal.Width （花瓣宽度），单位是 cm;
  - Labels：Iris Sentosa（山鸢尾）、 Iris Versicolor （杂色鸢尾），以及 Iris Virginica（维吉尼亚鸢尾）

补充材料：机器学习基本流程（python实现）

- 目标：建立模型，根据Feature预测Label，也即根据花的特征判断花的种类
- 流程：



补充材料：数据预处理

北京大学 PEKING UNIVERSITY

步骤1：使用pandas库

因为现在多数数据都可以用excel的.xlsx存储，所以只需要调用python的excel文件读写库pandas，然后直接读取数据即可。

在python中调用pandas库：**import pandas as pd**

使用：**Sample_excel=pd.read_excel(path, header = 0, index_col= 0)**可以直接调用path对应地址的excel文件，其中第0行为header行，第0列为index列，最后一列是label（默认）

步骤2：使用numpy库

因为之前一步得到Sample是带格式的，不是纯粹的数据，所以不能直接进行数学运算和模型生成，需要转成numpy的数组：**Sample = Sample_excel.values**

要想数据能够进行运算，一般都会调用numpy库：**import numpyas np**

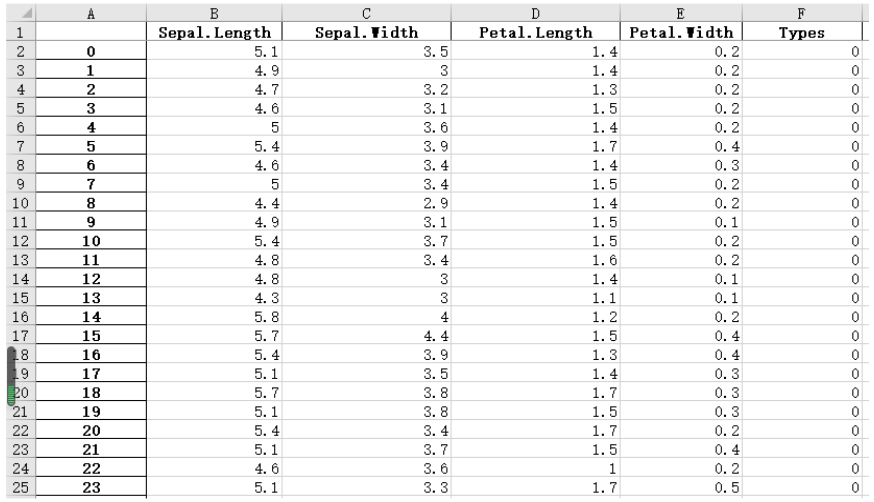
步骤3（不一定必须）：如果数据需要切割（例如区分Feature和Label），可以直接对numpy数组进行操作，例如：

y = Sample[:,Sample.shape[1]-1]

X = Sample[:,0:Sample.shape[1]-1]

补充材料：数据处理

- 经过刚才的步骤可将Excel表格中的数据集读入Python中
- 有些包可自动下载并准备数据集（如Pytorch的DataLoader），也可通过这些包进行数据集的准备
- 根据读取数据集文件的格式的不同，可选用不同的库进行读入：
  - 读取.gz文件使用gzip库
  - 读取二进制文件使用os库



	A	B	C	D	E	F
1		Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Types
2	0	5.1	3.5	1.4	0.2	0
3	1	4.9	3	1.4	0.2	0
4	2	4.7	3.2	1.3	0.2	0
5	3	4.6	3.1	1.5	0.2	0
6	4	5	3.6	1.4	0.2	0
7	5	5.4	3.9	1.7	0.4	0
8	6	4.6	3.4	1.4	0.3	0
9	7	5	3.4	1.5	0.2	0
10	8	4.4	2.9	1.4	0.2	0
11	9	4.9	3.1	1.5	0.1	0
12	10	5.4	3.7	1.5	0.2	0
13	11	4.8	3.4	1.6	0.2	0
14	12	4.8	3	1.4	0.1	0
15	13	4.3	3	1.1	0.1	0
16	14	5.8	4	1.2	0.2	0
17	15	5.7	4.4	1.5	0.4	0
18	16	5.4	3.9	1.3	0.4	0
19	17	5.1	3.5	1.4	0.3	0
20	18	5.7	3.8	1.7	0.3	0
21	19	5.1	3.8	1.5	0.3	0
22	20	5.4	3.4	1.7	0.2	0
23	21	5.1	3.7	1.5	0.4	0
24	22	4.6	3.6	1	0.2	0
25	23	5.1	3.3	1.7	0.5	0

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Types
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

可在Python中用Print命令打印数据集

补充材料：训练模型

北京大学 PEKING UNIVERSITY

- 机器学习的入门标准库： scikit learn（sklearn）
  - 英文版文档： https://scikit-learn.org
  - 中文版文档： https://www.scikitlearn.com.cn
  - 中文文档只是原理部分是中文的，函数文档部分还是英文的

- 标准的 **三步走**

模型选择根据任务不同而不同

  - 定义一个叫做 clf 的模型： **clf = neighbors.KNeighborsClassifier（15）**
  - 用训练数据训练 clf 模型： **clf.fit X_train,y_train**
  - 用测试数据进行预测，返回准确率： **accuracy = clf.score X_test,y_test**

补充材料：训练模型

- 对于稍微大点的数据集，一般都使用神经网络进行训练
- 所以会有专门用于搭建神经网络的库
- 例如Tensorflow和Pytorch
- 数据集以MNIST手写数字识别为例

- 60000个训练数据
- 10000个测试数据

补充材料：Tensorflow

北京大学 PEKING UNIVERSITY

- 神经网络的入门库1：tensorflow2.0
- 导入tensorflow2.0：import tensorflow as tf
- Tensorflow使用神经网络的三步走：

①创建一个神经网络

```
Def create_model(): 定义网络结构
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Flatten(input_shape= (28, 28)))
    model.add(tf.keras.layers.Dense(128, activation = 'relu'))
    model.add(tf.keras.layers.Dropout(0.2))
    model.add(tf.keras.layers.Dense(10, activation='softmax’))
    model.compile(optimizer='adam',loss='sparse_categorical_crossentropy', metrics=['accuracy']) #编译模型
    return model
model = create_model()
```

②用训练数据训练模型

```
model.fit(X_train, y_train, epochs = 5, batch_size= 32)
```

③用测试数据验证模型准确率

```
model.evaluate(X_test, y_test)
```

补充材料：Pytorch

北京大学 PEKING UNIVERSITY

- 神经网络的入门库2：pytorch
- tensorflow兼容性差，安装费劲，代码灵活程度低等因素，在学术圈和业界基本没有人用，学术圈一般用pytorch
- pytorch的**使用灵活**的同时，相应的**代码量也稍微大一点**

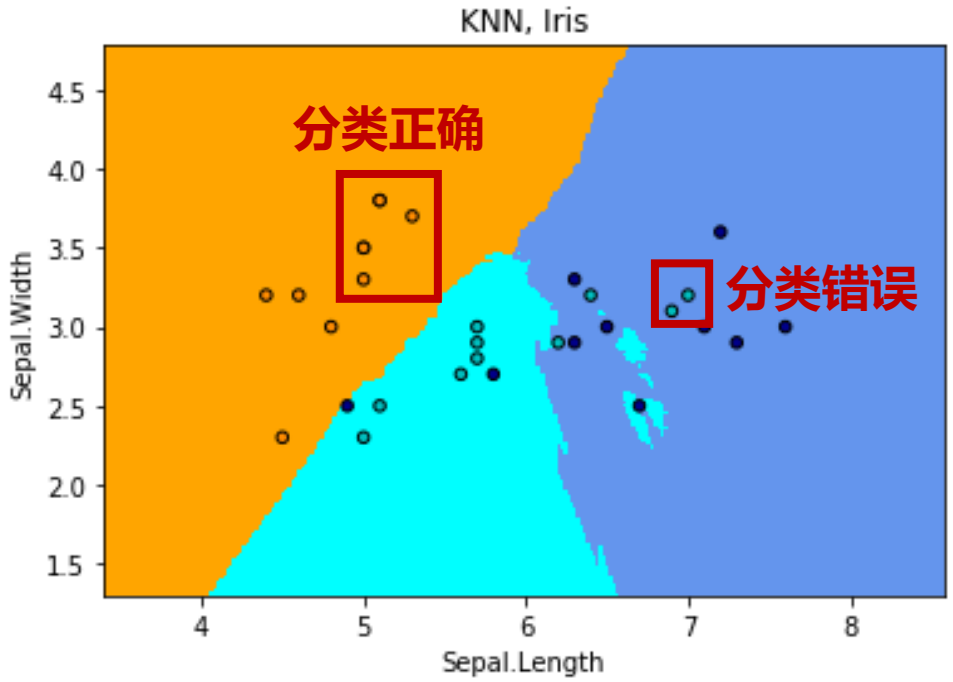
```
Train Epoch: 3 [44160/60000 (74%)]      Loss: 0.007931
Train Epoch: 3 [44800/60000 (75%)]      Loss: 0.099260
Train Epoch: 3 [45440/60000 (76%)]      Loss: 0.092140
Train Epoch: 3 [46080/60000 (77%)]      Loss: 0.168444
Train Epoch: 3 [46720/60000 (78%)]      Loss: 0.116179
Train Epoch: 3 [47360/60000 (79%)]      Loss: 0.092988
Train Epoch: 3 [48000/60000 (80%)]      Loss: 0.038248
Train Epoch: 3 [48640/60000 (81%)]      Loss: 0.019668
Train Epoch: 3 [49280/60000 (82%)]      Loss: 0.009166
Train Epoch: 3 [49920/60000 (83%)]      Loss: 0.095954
Train Epoch: 3 [50560/60000 (84%)]      Loss: 0.045420
Train Epoch: 3 [51200/60000 (85%)]      Loss: 0.161838
Train Epoch: 3 [51840/60000 (86%)]      Loss: 0.020038
Train Epoch: 3 [52480/60000 (87%)]      Loss: 0.020859
Train Epoch: 3 [53120/60000 (88%)]      Loss: 0.038819
Train Epoch: 3 [53760/60000 (90%)]      Loss: 0.060778
Train Epoch: 3 [54400/60000 (91%)]      Loss: 0.038955
Train Epoch: 3 [55040/60000 (92%)]      Loss: 0.020655
Train Epoch: 3 [55680/60000 (93%)]      Loss: 0.086007
Train Epoch: 3 [56320/60000 (94%)]      Loss: 0.025752
Train Epoch: 3 [56960/60000 (95%)]      Loss: 0.045303
Train Epoch: 3 [57600/60000 (96%)]      Loss: 0.092455
Train Epoch: 3 [58240/60000 (97%)]      Loss: 0.044798
Train Epoch: 3 [58880/60000 (98%)]      Loss: 0.023772
Train Epoch: 3 [59520/60000 (99%)]      Loss: 0.009263

Test set: Average loss: 0.0349, Accuracy: 9878/10000 (99%)
```

PyTorch

补充材料：验证模型结果与可视化

- 除了给出模型预测**准确率**数值之外，人们一般喜欢看图说话
- 所以最好有**直观的图能够直接表现出模型的好坏**
- 例如鸢尾花的例子

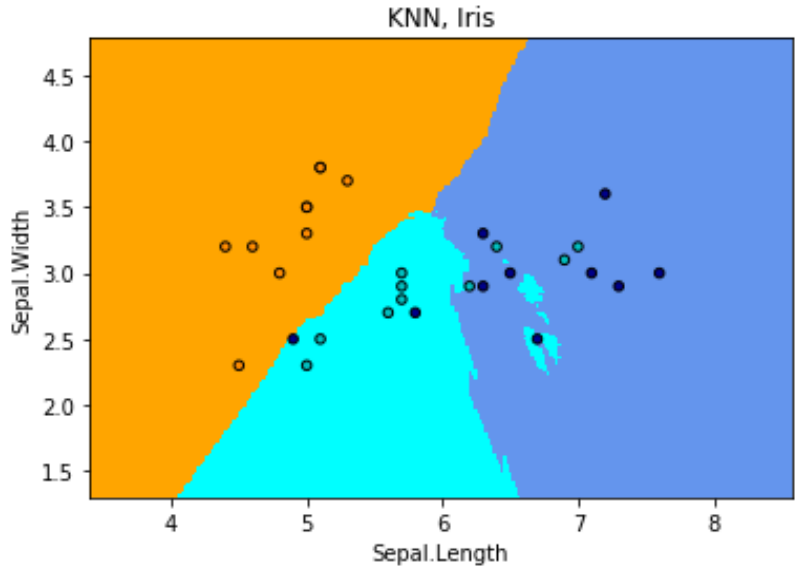


- 横坐标：花萼长度
- 纵坐标：花萼宽度
- 图像中的不同颜色区域/分界线：模型给出给出的分界线和各类区域
- 不同颜色的点，代表测试数据中每个个体处在的实际位置，同颜色的点调入同色区域内代表正确，否则错误。

补充材料：Matplotlib

- 使用matplotlib库进行科学画图
- 导入matplotlib的一般绘图模块pyplot：import matplotlib.pyplotas plt
- 导入matplotlib染色模块：from matplotlib.colorsimport ListedColormap
- 使用matplotlib进行画图：

```
plt.figure(2) #第几张图
plt.pcolormesh(xx, yy, y_predict, cmap=cmap_light)
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold, edgecolor='k', s=20)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xlabel("Sepal.Length")
plt.ylabel("Sepal.Width")
plt.title("KNN, Iris")
```

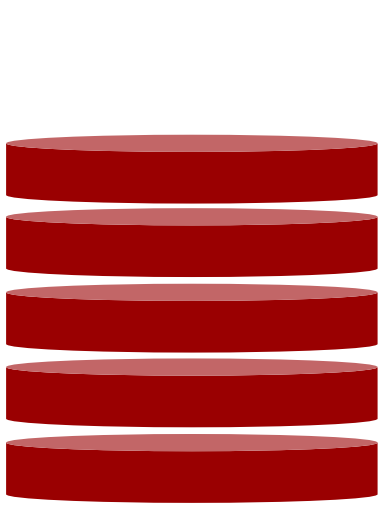


补充材料：Matplotlib

- 科学绘图库matplotlib
- 官网文档：https://matplotlib.org/
- 几乎所有形式的科学制图都可以从example中找到

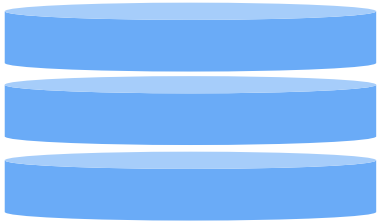
补充材料：总结

- 可运用Python中的各种库完成机器学习任务



数据收集预处理

测试集



训练集

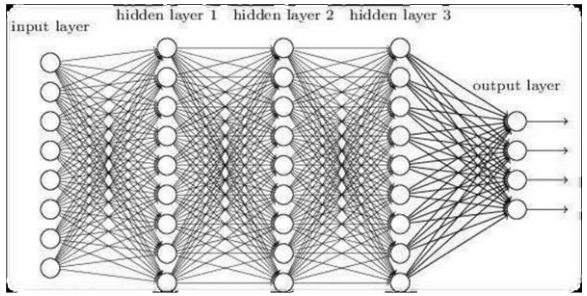
①将数据集导入python

Pandas+numpy+…

③验证模型与可视化

在测试集上评估模型性能

matplotlib



②用训练集训练模型

Sklearn+tensorflow+pytorch

谢　谢

Thank You!