# 2021 年夏季学期 Python 程序设计大作业指引

# 人脸性别识别——二分类问题

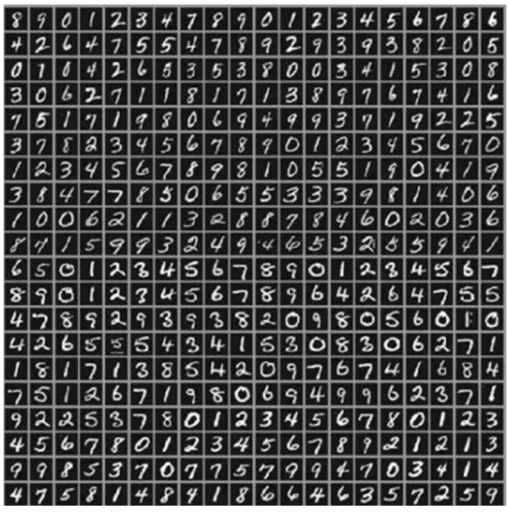
分类问题是机器学习的一个重要基础问题。分类问题的目标可以描述为，根据已知样本的某些特征，判断一个新的样本属于哪种已知的样本类。正因如此，绝大多数的分类问题被认为属于有监督学习，从样本和标签中学习分类模型。

机器学习里最常用的分类算法主要有以下几种:

- 线性分类器
    - 线性判别分析（LDA）
    - 逻辑回归（logistic regression）
    - 朴素贝叶斯分类器（naive bayes classifier）
    - 感知器（perceptron）
- 支持向量机（support vector machine, SVM）
    - 最小二乘支持向量机（least squares support vector machines ）
- 二次分类器（quadratic classifier）
- 核估计（kernel estimation）
    - K 近邻法（k-nearest neighbor, KNN）
- Boosting 算法
    - 梯度增强（Gradient Boosting）
    - 自适应增强（Adaboost）
- 决策树（decision trees）
    - 随机森林（random forests）
- 神经网络（neural networks）
- 学习式向量量化（learning vector quantization）

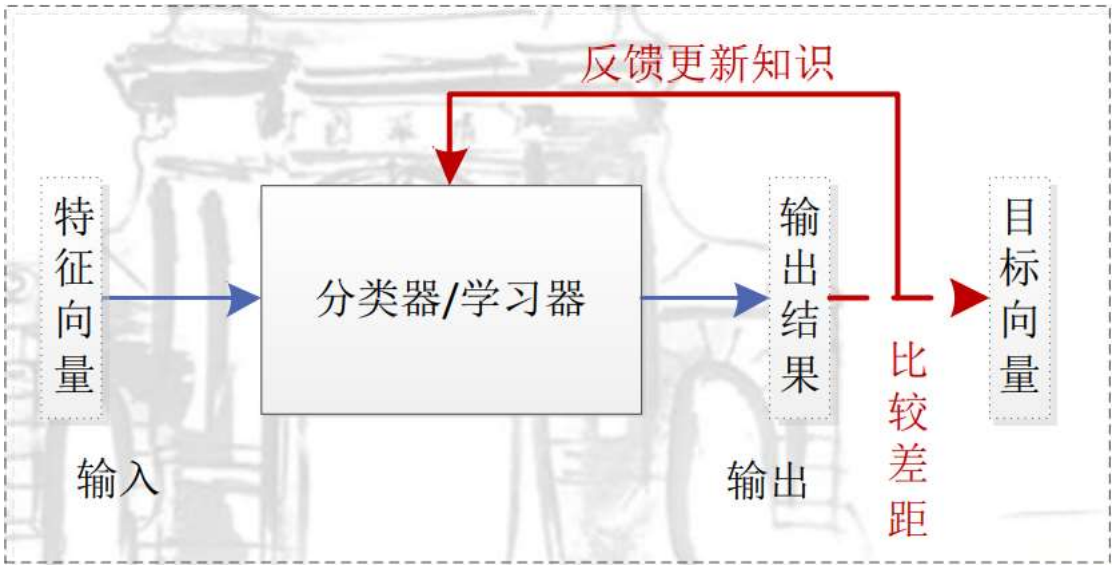
在本次课程当中，已经介绍了多种基本方法，下面以手写数字 0-9 的十分类判别问题为例，简单介绍图像分类问题的处理。

**MNIST 手写数字十分类问题**



如图所示，手写数字识别问题一个经典数据集即为来自美国国家标准与技术研究所（National Institute of Standards and Technology, NIST）的 MNIST 数据集。它提供了由 250 个不同人员手写的数字，供使用者训练和测试手写数字识别模型。

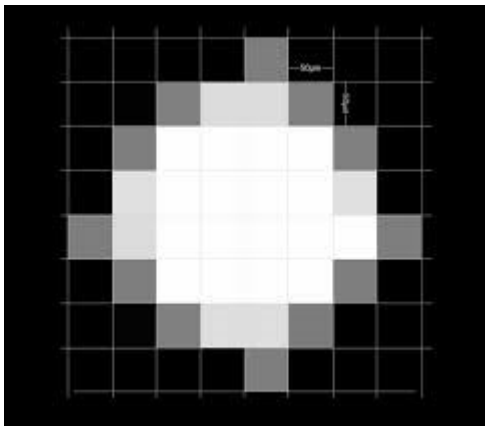
在附件中，我们给出了 3 种不同路线训练手写数字识别模型的代码：K 近邻、逻辑回归和 CNN。CNN 供学有余力的同学自学了解。K 近邻和逻辑回归两种方法分别提供两个不同的实现，即调用了 sklearn 库与非 sklearn 的实现方法。




我们的代码遵循如图所示的模型训练基本流程，编写代码的助教已经在文件中附上了适当的注释方便理解。下面，就一些细节问题进行说明：

**输入。**MNIST 数据集所提供的每“条”数据是一张 28*28 的灰度图像（即图像是黑白的）。所谓 28*28，是指图像是由像素点构成，按照水平方向 28 列、垂直方向 28 行排列。

如果将灰度图像的每个像素理解为“点”，那么整幅图像就是由点构成的点阵。如图所示，每个“点”具备一个灰度值，是取值范围[0,255]的整数。所以，一张灰度图像可以很自然地理解为一个矩阵，矩阵的每个元素都是取值范围[0,255]的整数。



*图片来自网络*

为了令同学们将注意力集中于模型训练代码本身，我们在附件的样例代码中额外配备有 DataLoader.py，它是封装好的用于读入 MNIST 数据集的代码，在完成最终大作业时，在数据输入环节，可以参考这个文件的内容。

MNIST 数据集的标签采用了一种独特的 one-hot 编码，尽管最终大作业的数据集并没

有采用这个编码，但为了方便同学理解样例代码，在此提供一篇参考文《详解 One Hot 编码 – 附代码》。简单来说，如果一张图片的分类是“1”，那么其 one-hot 编码就是向量[0,1,0,0,0,0,0,0,0,0]。

**数据集划分。**一般而言，需要将数据集划分为训练集、验证集和测试集使用。训练集用于训练模型，验证集用于在迭代优化过程中验证模型，测试集用于最终检验模型效果。如果将全部数据都用于训练，那么可能会得到一个与训练数据高度契合（准确率甚至可以达到100%），但用于真实世界的普通数据效果却非常差的模型。这称之为机器学习的*过拟合*。因此，需要验证集和测试集来改进、评价我们的模型。

有一种简略做法是不设验证集，而使用测试集进行验证工作，因此也有主张将数据集划分两部分就好的。

在我们的示例 DataLoader.py 中，训练集、验证集和测试集的划分比例被设置为了 65%、20%和 15%。

$$trainIndex = int(self.dataCount * 10 * 0.65)$$

$$validationIndex = int(self.dataCount * 10 * 0.85)$$

这就是最简单的“留出法”划分。除此之外还有留一法和 K 折交叉验证，学有余力的同学可以自行了解。

**模型训练。**请参考样例代码中的注释并结合课件进行理解。

**模型评价。**在评估所得到的模型时，我们需要一些适当的指标来辅助。机器学习最常用的评价标准就是模型的*混淆矩阵*。

True Positive（真阳，TP）：将阳性样本预测为阳性。

True Negative（真阴，TN）：将阴性预测为阴性。

False Positive（假阳，FP）：将阴性预测为阳性数，即误报（Type I error）。

False Negative（假阴，FN）：将阳性预测为阴性数，即漏报（Type II error）。

可能有些同学会觉得“假阳性”“假阴性”这一名词十分熟悉，是的，混淆矩阵最常用的一个场景就是医学药物及检测的效果评估。而且，在概率论及数理统计初步中，我们还将学习 Type I 和 Type II 错误的更详细的知识。机器学习与统计学有十分紧密的联系，有志于机器学习方向的同学务必扎牢统计学知识的基础。

更进一步，在此四类状况之上有一些“率”和“度”的指标：

1. 灵敏度（sensitive）

$$sensitive = TP/(TP + FN)$$

即“真阳性率”。

2. 特效度（specificity）

$$specificity = TN/(TN + FP)$$

即“真阴性率”。

3. 精确率、精度（precision）

$$precision = TP/(TP + FP)$$

4. 准确率（accuracy）

$$accuracy = (TP + TN)/(TP + TN + FP + FN)$$

5. 错误率（error rate）

$$error\_rate = (FP + FN)/(TP + TN + FP + FN)$$

6. 召回率（recall）

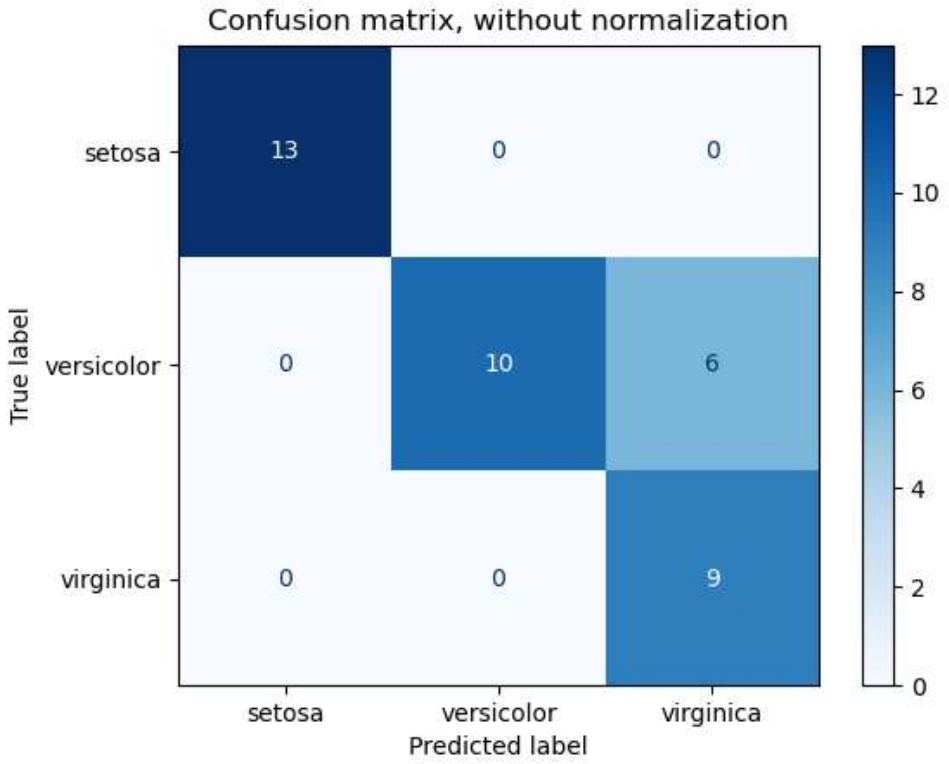
$$recall = TP/(TP + FN)$$

注意，有些资料错误地认为召回率与精确率等同，实际上二者的分母有本质差别。灵敏度是“所有被判为阳性的样本数”，召回率是“所有真正的阳性样本数”。但召回率与灵敏度的公式是相同的，在不同学科领域的文献中各有使用。

除此之外，还有一些机器学习的曲线图指标，在此仅建议学有余力的同学了解。

**分类问题模型评估的可视化。**为了更直观地观看模型的效果，有一种将混淆矩阵可视化的方法，在学术写作中被广泛使用。

如图所示的混淆矩阵，横轴代表预测结果分类，纵轴代表样本的真实分类，将每种类型状况的累积样本数用于上色绘制（注意，需要对样本数进行归一化然后换算为填充色明暗度0-255）。可以自主设定颜色越明亮数字越高，或与之相反。




*（考虑到引用的个人创作内容的知识产权问题，在此采用了 scikit-learn.org 配图，MNIST 的混淆矩阵可视化有很多样例，可以自行搜索）*

**大作业——人脸性别识别二分类问题**

在本次大作业中，我们提供了 LFW (Labeled Faces in the Wild) 人脸数据集，这是最常用的人脸识别数据集之一。它包含多个不同自然人的人脸彩色照片，每张尺寸为 250*250，同一个人可能具有不同角度的 2 张照片（正视、侧视）。要求，训练人脸识别模型，输入一张测试人脸照片（同样来自 LFW 数据集），能够预测照片中人士的性别。

为了简化问题的难度，在本次作业中，我们仅考虑二分类性别，即男性（male）和女性（female）。请同学参考 MNIST 十分类问题的样例代码，编写自己的模型并进行训练，以及给出最终的测试结果和结果评估。

**输入数据的处理提示。**LFW 数据集的图片绝大多数是彩色图片，彩色图片的每个像素不再是单一的灰度值（熟悉 Photoshop 等图像处理软件的同学应该立刻能够联想到黑白通道

及彩色通道等名词），而是多个色彩值。一般而言，不带透明通道的彩色图片是 RGB 三个彩色通道。亦即，此时读入一张图片，每个像素有 3 个不同色彩通道值。那么，原有的二维矩阵就拓展到了三维。

图片的读入在 python 中有库函数可以提供支持，譬如我们所提供的数据读入模块 DataLoader.py 采用了 PIL 库。

对于性别判定问题，灰度图像已经基本够用。因此在处理时，可以将输入图像转换一下再进行训练。对于 PIL 库而言，可以使用一个简单的 convert()函数将彩色图片转换为灰度图。

```
from PIL import Image

Img = Image.open('example.png')
ImgGray = Img.convert('L')    #转化为灰度图
```

**标签处理。**本次数据集的每张图像都有其文件名，数据标签以两个 txt 文档形式给出。分别是 male_names 和 female_names。一种简单的读入办法是，将标签值存放在 dict 对象中，键（key）为文件名，值（value）为性别编码。同学也可以自主思考更加快捷有效率的处理方法。

**放轻松！**什么样的训练模型可以被认为是在数据上“起作用”了？一种简单的思考方法是，只要分类的准确率不低于样本的分布就可以。也就是说，对二分类问题，分类器至少应该有 50%的准确率，否则就不如随机猜测了。

实际上，对于二分类问题，人脸识别数据集在简单的模型上也有很好的表现，只要你选择了适当的训练终点。本次作业对准确率没有特殊要求，请同学们不要过于紧张。熟悉 Python 语法以及 Python 在实际背景中的应用才是作业所期望的重点。