



北京师范大学 珠海校区
BEIJING NORMAL UNIVERSITY AT ZHUHAI

Python安装和使用

马静



We can read of things that happened
5,000 years ago in the Near East,
where people first learned to write.
But there are some parts of the world
where even now people cannot write.

CONTENT

- 01 安装
- 02 Sklearn学习
- 03 Sklearn实践
- 04 Kaggle的一个分类器的实践



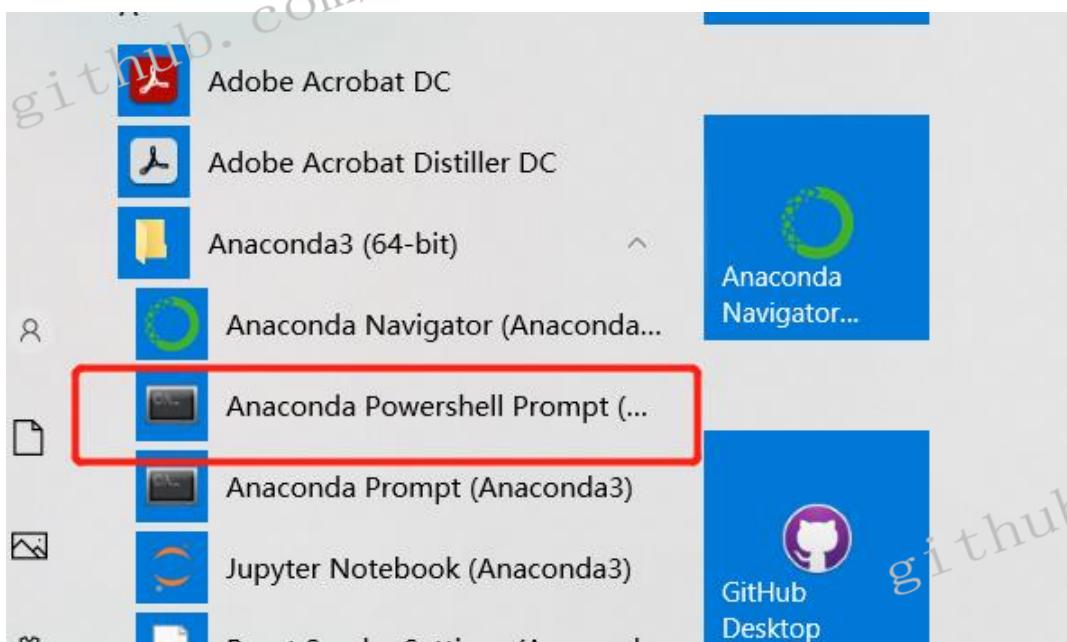
北京師範大學 珠海校區
BEIJING NORMAL UNIVERSITY AT ZHUHAI



01 安装



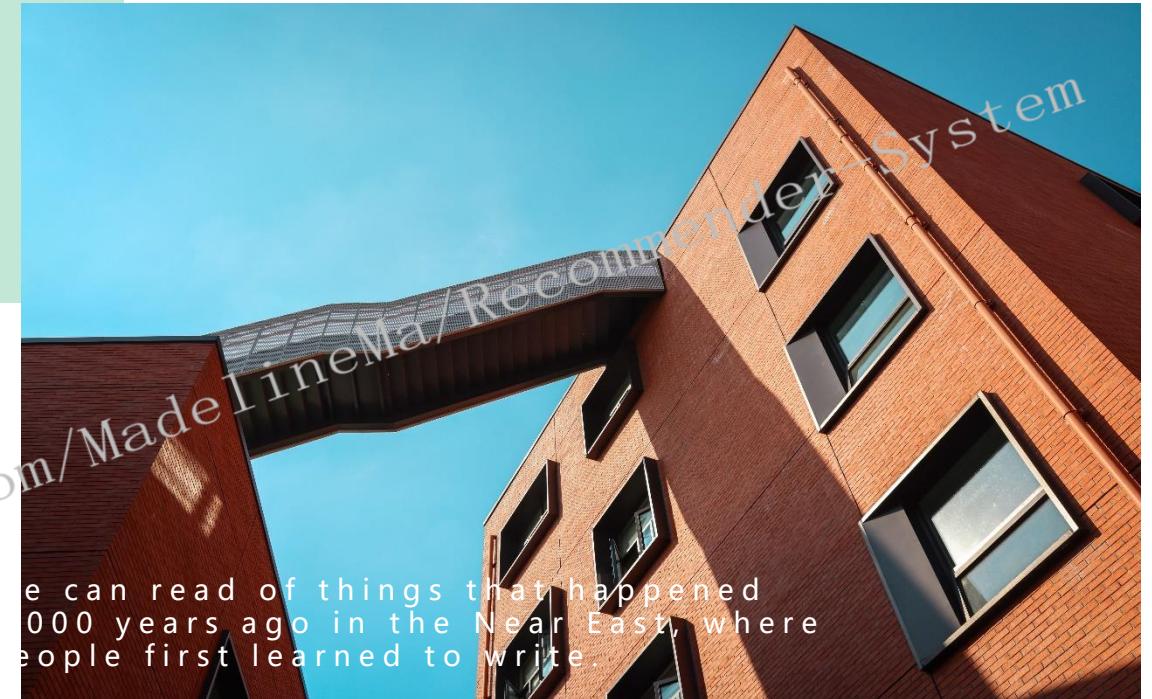
[【Anaconda教程01】怎么安装Anaconda3 - 知乎 \(zhihu.com\)](#)



Sklearn的安装
pip install scikit-learn



02 Sklearn學習





机器学习分类

机器学习一般分为下面几种类别

- 监督学习 (Supervised Learning)
- 无监督学习 (Unsupervised Learning)
- 强化学习 (Reinforcement Learning , 增强学习)
- 半监督学习 (Semi-supervised Learning)
- 深度学习 (Deep Learning)



Python Scikit-learn

- <http://scikit-learn.org/stable/>
 - Machine Learning in Python
 - 一组简单有效的工具集
 - 依赖Python的NumPy , SciPy和matplotlib库
 - 开源、可复用

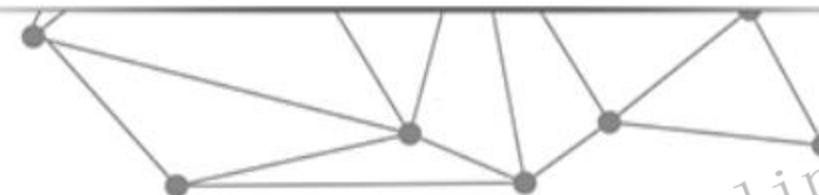


Scikit-learn 常用函数

	应用 (Applications)	算法 (Algorithm)
分类 (Classification)	异常检测，图像识别，等	KNN, SVM , etc.
聚类 (Clustering)	图像分割，群体划分，等	K-Means , 谱聚类, etc.
回归 (Regression)	价格预测，趋势预测，等	线性回归，SVR , etc.
降维 (Dimension Reduction)	可视化，	PCA , NMF , etc.



sklearn库中的标准数据集





数据集总览

	数据集名称	调用方式	适用算法	数据规模
小数据集	波士顿房价数据集	load_boston()	回归	506*13
	鸢尾花数据集	load_iris()	分类	150*4
	糖尿病数据集	load_diabetes()	回归	442*10
	手写数字数据集	load_digits()	分类	5620*64
大数据集	Olivetti 脸部图像数据集	fetch_olivetti_faces()	降维	400*64*64
	新闻分类数据集	fetch_20newsgroups()	分类	-
	带标签的人脸数据集	fetch_lfw_people()	分类；降维	-
	路透社新闻语料数据集	fetch_rcv1()	分类	804414*47236

注：小数据集可以直接使用，大数据集要在调用时程序自动下载（一次即可）。



波士顿房价数据集

波士顿房价数据集包含506组数据，每条数据包含房屋以及房屋周围的详细信息。其中包括城镇犯罪率、一氧化氮浓度、住宅平均房间数、到中心区域的加权距离以及自住房平均房价等。因此，波士顿房价数据集能够应用到回归问题上。



波士顿房价数据集

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4

图. 部分房价数据



波士顿房价数据集-属性描述

CRIM : 城镇人均犯罪率。

ZN : 住宅用地超过 25000 sq.ft. 的比例。

INDUS : 城镇非零售商用土地的比例。

CHAS : 查理斯河空变量 (如果边界是河流，则为1；否则为0)

NOX : 一氧化氮浓度。

RM : 住宅平均房间数。

AGE : 1940 年之前建成的自用房屋比例。

DIS : 到波士顿五个中心区域的加权距离。

RAD : 辐射性公路的接近指数。

TAX : 每 10000 美元的全值财产税率。

PTRATIO : 城镇师生比例。

B : $1000 (Bk - 0.63)^2$, 其中 Bk 指代城镇中黑人的比例。

LSTAT : 人口中地位低下者的比例。

MEDV : 自住房的平均房价，以千美元计。



波士顿房价数据集

使用`sklearn.datasets.load_boston`即可加载相关数据集

其重要参数为：

- `return_X_y`: 表示是否返回target (即价格) , 默认为False , 只返回data (即属性) 。



波士顿房价数据集-加载示例

示例1：

```
>>> from sklearn.datasets import load_boston
>>> boston = load_boston()
>>> print(boston.data.shape)
(506, 13)
```

示例2：

```
>>> from sklearn.datasets import load_boston
>>> data, target = load_boston(return_X_y=True)
>>> print(data.shape)
(506, 13)
>>> print(target.shape)
(506)
```



鸢尾花数据集

鸢尾花数据集采集的是鸢尾花的测量数据以及其所属的类别。

测量数据包括：萼片长度、萼片宽度、花瓣长度、花瓣宽度。

类别共分为三类：Iris Setosa , Iris Versicolour , Iris Virginica。该数据集可用于多分类问题。

萼片长度	萼片宽度	花瓣长度	花瓣宽度	类别
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa
4.8	3	1.4	0.1	Iris-setosa
4.3	3	1.1	0.1	Iris-setosa
5.8	4	1.2	0.2	Iris-setosa

图. 鸢尾花数据集分数据示例



鸢尾花数据集

使用sklearn.datasets. **load_iris**即可加载相关数据集

其参数有：

- **return_X_y**:若为True，则以 (data, target) 形式返回数据；默认为False，表示以字典形式返回数据全部信息（包括 data和target）。



鸢尾花数据集-加载示例

示例：

```
>>> from sklearn.datasets import load_iris
>>> iris = load_iris()
>>> print(iris.data.shape)
(150, 4)
>>> print(iris.target.shape)
(150, )
>>> list(iris.target_names)
['setosa', 'versicolor', 'virginica']
```



手写数字数据集

手写数字数据集包括1797个0-9的手写数字数据，每个数字由 8×8 大小的矩阵构成，矩阵中值的范围是0-16，代表颜色的深度。



手写数字数据集

0	0	5	13	9	1	0	0
0	0	13	15	10	15	5	0
0	3	15	2	0	11	8	0
0	4	12	0	0	8	8	0
0	5	8	0	0	9	8	0
0	4	11	0	1	12	7	0
0	2	14	5	10	12	0	0
0	0	6	13	10	0	0	0



图. 数字0的样本



手写数字数据集

使用`sklearn.datasets.load_digits`即可加载相关数据集

其参数包括：

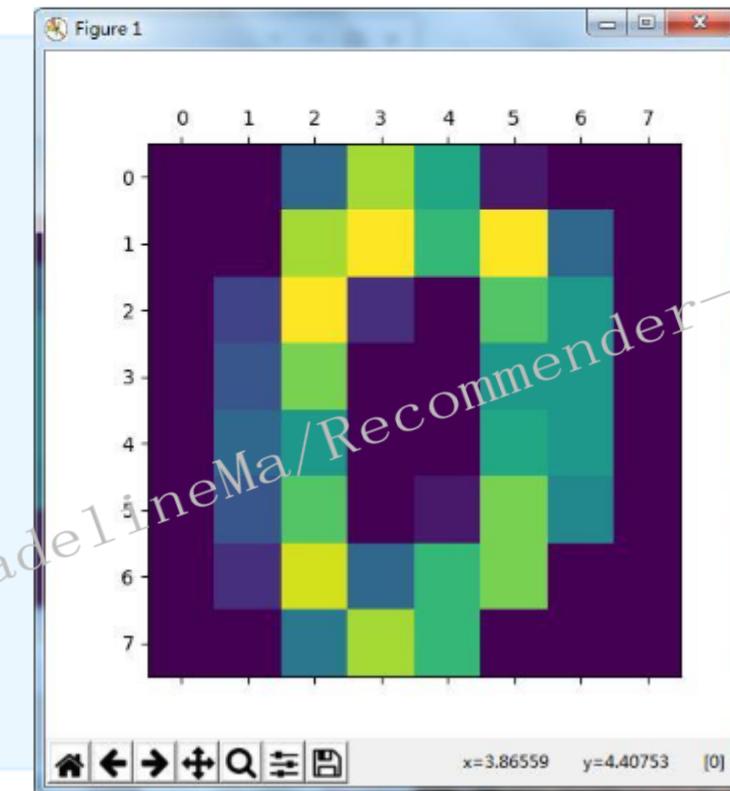
- `return_X_y`: 若为True，则以 (data, target) 形式返回数据；默认为False，表示以字典形式返回数据全部信息（包括data和target）；
- `n_class` : 表示返回数据的类别数，如：`n_class=5`, 则返回0到4的数据样本。



手写数字数据集

示例：

```
>>> from sklearn.datasets import load_digits
>>> digits = load_digits()
>>> print(digits.data.shape)
(1797, 64)
>>> print(digits.target.shape)
(1797, )
>>> print(digits.images.shape)
(1797, 8, 8)
>>> import matplotlib.pyplot as plt
>>> plt.matshow(digits.images[0])
>>> plt.show()
```





sklearn库的基本功能



sklearn库的基本功能

sklearn库的共分为6大部分，分别用于完成分类任务、回归任务、聚类任务、降维任务、模型选择以及数据的预处理。



分类任务

分类模型	加载模块
最近邻算法	neighbors.NearestNeighbors
支持向量机	svm.SVC
朴素贝叶斯	naive_bayes.GaussianNB
决策树	tree.DecisionTreeClassifier
集成方法	ensemble.BaggingClassifier
神经网络	neural_network.MLPClassifier



03 Sklearn实战



We can read of things that
happened 5,000 years ago in
the Near East, where people
first learned to write.





- 二分类问题
- 调用LR模型对蘑菇是否有毒进行预测
- 数据来源：Kaggle
- [Mushroom Classification | Kaggle](#)
- 代码：git/exercises/sk_lr_mushroom.py

Mushroom Classification
UCI Machine Learning - Updated 4 years ago
Usability 8.5 · 1 File (CSV) · 34 KB · 3 Tasks

◀ mushrooms.csv (365.24 KB)

Detail Compact Column 10 of 23 columns ▾

About this file

Attribute Information: (classes: edible=e, poisonous=p)

- cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
- cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
- cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y
- bruises: bruises=t,no=f

class	cap-shape	cap-surface	cap-color	bruises
edible=e, poisonous=p	bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s	fibrous=f,grooves=g,scaly=y,smooth=s	brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y	bruises=t,no=f
e	52%	y	40%	n
p	48%	s	31%	g
	Other (1316)	16%	Other (2324)	Other (4000)
		29%	29%	49%
				true 0 0%
				false 0 0%



- 多分类问题
- 调用MLP模型对手写体进行分类
- 数据源：UCI
<http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits>
- 代码: git/exercises/sk_mlp_mushroom.py
- 讲解的手写体demo数据集有待处理，希望感兴趣学生可帮忙完成

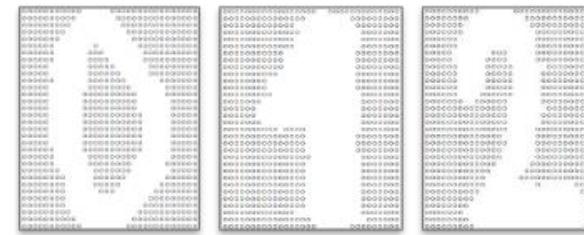
The screenshot shows the UCI Machine Learning Repository homepage. At the top, there is a logo featuring a yellow 'UCI' monogram and a blue illustration of a hand holding a pen. Below the logo, the text 'Machine Learning Repository' and 'Center for Machine Learning and Intelligent Systems' is displayed. A brown banner below the header contains the text: 'Check out the [beta version](#) of the new UCI Machine Learning Repository we are currently testing! site.' The main content area features a section titled 'Pen-Based Recognition of Handwritten Digits Data Set'. It includes a 'Download' link pointing to 'Data Folder' and 'Data Set Description'. An 'Abstract' section states: 'Digit database of 250 samples from 44 writers'. Below the abstract is a table with the following data:

Data Set Characteristics:	Multivariate	Number of Instances:	10992	Area:	Computer
Attribute Characteristics:	Integer	Number of Attributes:	16	Date Donated	1998-07-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	248643



任务介绍

手写数字识别是一个多分类问题，共有10个分类，每个手写数字图像的类别标签是0~9中的其中一个数。例如下面这三张图片的标签分别是0, 1, 2。

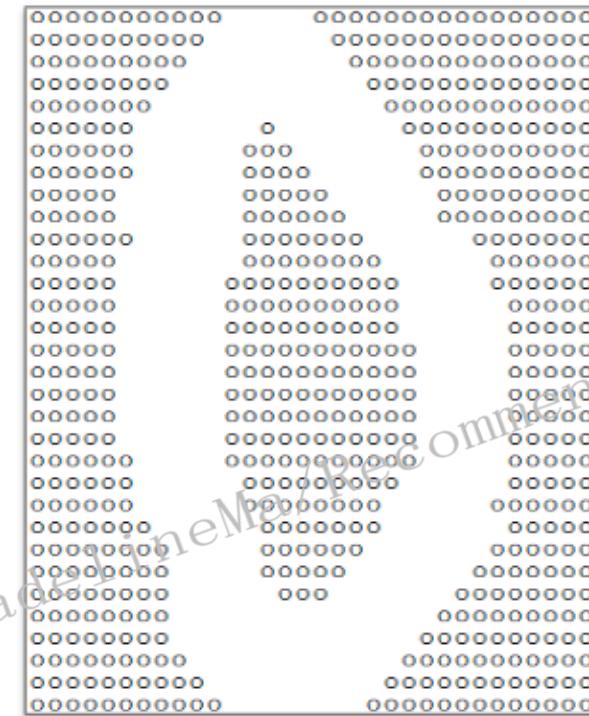


任务：利用sklearn来训练一个简单的全连接神经网络，即多层感知机（Multilayer perceptron, MLP）用于识别数据集DBRHD的手写数字。



MLP的输入

- DBRHD数据集的每个图片是一个由0或1组成的 32×32 的文本矩阵；
- 多层感知机的输入为图片矩阵展开的 1×1024 个神经元。





MLP的输出

MLP输出：“one-hot vectors”

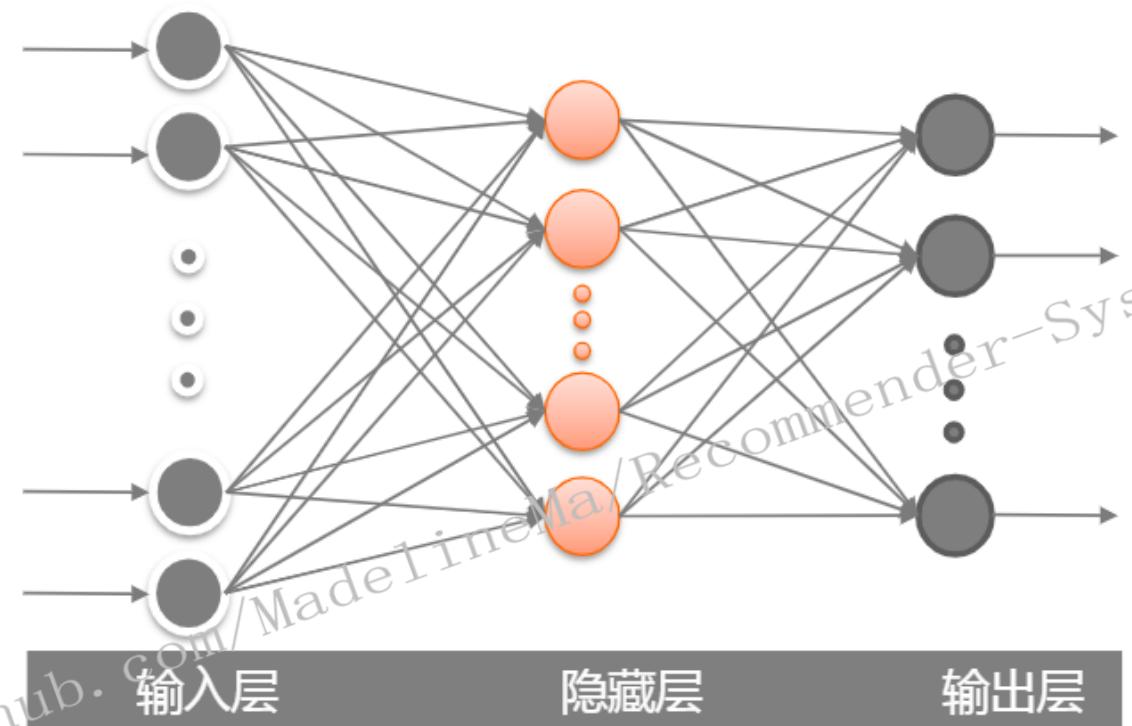
- 一个one-hot向量除了某一位的数字是1以外其余各维度数字都是0。
- 图片标签将表示成一个只有在第n维度（从0开始）数字为1的10维向量。

比如，标签0将表示成 $[1,0,0,0,0,0,0,0,0,0]$ 。即，MLP输出层具有10个神经元。



MLP结构

- MLP的输入与输出层，中间隐藏层的层数和神经元的个数设置都将影响该MLP模型的准确率。
- 在本实例中，我们只设置一层隐藏层，在后续实验中比较该隐藏层神经元个数为50、100、200时的MLP效果。





MLP手写识别实例构建

本实例的构建步骤如下：

- 步骤1：建立工程并导入sklearn包
- 步骤2：加载训练数据
- 步骤3：训练神经网络
- 步骤4：测试集评价



步骤1：建立工程并导入sklearn包

1) 创建sklearnBP.py文件

2) 在sklearnBP.py文件中导入sklearn相关包

```
import numpy as np      #导入numpy工具包
from os import listdir #使用listdir模块，用于访问本地文件
from sklearn.neural_network import MLPClassifier
```



步骤2：加载训练数据

1) 在sklearnBP.py文件中，定义img2vector函数，将加载的32*32的图片矩阵展开成一列向量

```
def img2vector(fileName):
    retMat = np.zeros([1024],int) #定义返回的矩阵，大小为1*1024
    fr = open(fileName)           #打开包含32*32大小的数字文件
    lines = fr.readlines()        #读取文件的所有行
    for i in range(32):          #遍历文件所有行
        for j in range(32):      #并将01数字存放在retMat中
            retMat[i*32+j] = lines[i][j]
    return retMat
```



步骤2：加载训练数据

2) 在sklearnBP.py文件中定义加载训练数据的函数readDataSet ,
并将样本标签转化为one-hot向量

```
def readDataSet(path):
    fileList =.listdir(path)      #获取文件夹下的所有文件
    numFiles = len(fileList)       #统计需要读取的文件的数目
    dataSet = np.zeros([numFiles,1024],int) #用于存放所有的数字文件
    hwLabels = np.zeros([numFiles,10])      #用于存放对应的标签one-hot
    for i in range(numFiles):    #遍历所有的文件
        filePath = fileList[i]    #获取文件名称\路径
        digit = int(filePath.split('.')[0]) #通过文件名获取标签
        hwLabels[i][digit] = 1.0.      #将对应的one-hot标签置1
        dataSet[i] = img2vector(path +'/' +filePath) #读取文件内容
    return dataSet,hwLabels
```



步骤3：训练神经网络

1) 在sklearnBP.py文件中 构建神经网络：设置网络的隐藏层数、各隐藏层神经元个数、激活函数、学习率、优化方法、最大迭代次数。

- 设置含100个神经元的隐藏层。
- hidden_layer_sizes 存放的是一个元组，表示第i层隐藏层里神经元的个数
- 使用logistic激活函数和adam优化方法，并令初始学习率为0.0001，

```
clf = MLPClassifier(hidden_layer_sizes=(100,),  
                     activation='logistic', solver='adam',  
                     learning_rate_init = 0.0001, max_iter=2000)
```



步骤3：训练神经网络

1) 在sklearnBP.py文件中，构建神经网络：设置网络的隐藏层数、各隐藏层神经元个数、激活函数、学习率、优化方法、最大迭代次数。

- 设置含100个神经元的隐藏层。
- hidden_layer_sizes 存放的是一个元组，表示第i层隐藏层里神经元的个数
- 使用logistic激活函数和adam优化方法，并令初始学习率为0.0001，

```
clf = MLPClassifier(hidden_layer_sizes=(100,),  
                     activation='logistic', solver='adam',  
                     learning_rate_init = 0.0001, max_iter=2000)
```



步骤3：训练神经网络

2) 在sklearnBP.py文件中，使用训练数据训练构建好的神经网络

- fit函数能够根据训练集及对应标签集自动设置多层感知机的输入与输出层的神经元个数。
- 例如train_dataSet为n*1024的矩阵，train_hwLabels为n*10的矩阵，则fit函数将MLP的输入层神经元个数设为1024，输出层神经元个数为10：

```
clf.fit(train_dataSet,train_hwLabels)
```



步骤4：测试集评价

1) 在sklearnBP.py文件中，加载测试集

```
dataSet, hwLabels = readDataSet('testDigits')
```

2) 使用训练好的MLP对测试集进行预测，并计算错误率：

```
res = clf.predict(dataSet)          #对测试集进行预测
error_num = 0                      #统计预测错误的数目
num = len(dataSet)                 #测试集的数目
for i in range(num):               #遍历预测结果
    #比较长度为10的数组，返回包含01的数组，0为不同，1为相同
    #若预测结果与真实结果相同，则10个数字全为1，否则不全为1
    if np.sum(res[i] == hwLabels[i]) < 10:
        error_num += 1
print("Total num:", num, " Wrong num:", \
      error_num, " WrongRate:", error_num / float(num))
```



实验效果

隐藏层神经元个数影响

运行隐藏层神经元个数为50、100、200的多层感知机，对比实验效果：

神经元个数	50	100	200
错误数量	47	40	37
正确率	0. 9503	0. 9577	0. 9608

- 随着隐藏层神经元个数的增加，MLP的正确率持上升趋势；
- 大量的隐藏层神经元带来的计算负担与对结果的提升并不对等，因此，如何选取合适的隐藏神经元个数是一个值得探讨的问题。



实验效果

迭代次数影响分析:

我们设隐藏层神经元个数为100，初始学习率为0.0001，最大迭代次数分别为500、1000、1500、2000，结果如下：

学习率	500	1000	1500	2000
错误数量	50	41	41	40
正确率	0. 9471	0. 9567	0. 9567	0. 9577

- 过小的迭代次数可能使得MLP早停，造成较低的正确率。
- 当最大迭代次数>1000时，正确率基本保持不变，这说明MLP在第1000迭代时已收敛，剩余的迭代次数不再进行。



实验效果

学习率影响分析：

改用随机梯度下降优化算法即将MLPclassifier的参数（`solver='sgd'`，），设隐藏层神经元个数为100，最大迭代次数为2000，学习率分别为：0.1、0.01、0.001、0.0001，结果如下：

学习率	0.1	0.01	0.001	0.0001
错误数量	35	41	49	222
正确率	0.9630	0.9567	0.9482	0.7653

结论：较小的学习率带来了更低的正确率，这是因为较小学习率无法在2000次迭代内完成收敛，而步长较大的学习率使得MLP在2000次迭代内快速收敛到最优解。因此，较小的学习率一般要配备较大的迭代次数以保证其收敛。



本周作业

1. 熟悉Python自带数据资源以及第一节课提供的公共资源；
2. 尝试自己写一个LR的算法，或网上找一个，代替Sklearn库算法进行demo1的训练。





北京師範大學 珠海校區
BEIJING NORMAL UNIVERSITY AT ZHUHAI

THANKS

DESIGNED BY 2xh