**机器学习课程报告**

**CNN**

**一、CNN任务描述**

基于 MNIST 数据集，**设计**并**实现**一个合适的卷积神经网络。

**二、CNN实验准备**

**（一）导入必要的包**

keras以TensorFlow和Theano作为后端封装，是一个专门用于深度学习的python模块。

包含了全连接层，卷积层，池化层，循环层，嵌入层等等等，常见的深度学习模型。

包含用于定义损失函数的Losses，用于训练模型的Optimizers，评估模型的Metrics，定义激活函数的Activations，防止过拟合的Regularizers等功能

import numpy as np  
import pandas as pd  
  
np.random.seed(0)  
import random  
  
import matplotlib.pyplot as plt  
  
from keras.models import Sequential  
from keras.layers import Dense,Dropout,Convolution2D,MaxPooling2D,Flatten  
from keras.optimizers import Adam  
from keras.datasets import mnist  
from keras.utils import to\_categorical

**（二）使用的layer说明**

**1.Dense（维度不变）**

全连接层，只对数据最后一个维度进行处理。

**2.Flatten**

Flatten层用来将输入“压平”，即把多维的输入一维化，常用在从卷积层到全连接层的过渡。Flatten不影响batch的大小。

**3.Conv2D**

二维卷积层，即对图像的空域卷积。

**4.Dropout**

为输入数据施加Dropout。Dropout将在训练过程中每次更新参数时随机断开一定百分比（rate）的输入神经元，Dropout层用于防止过拟合。

**5.MaxPooling2D**

空间池化（也叫亚采样或下采样）降低了每个特征映射的维度，但是保留了最重要的信息。空间池化可以有很多种形式：最大(Max)，平均(Average)，求和(Sum)等等。

**三、MNIST数据集描述**

**（一）导入MNIST 数据集**

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

一共统计了来自250个不同的人手写数字图片，其中50%是高中生，50%来自人口普查局的工作人员。该数据集的收集目的是希望通过算法，实现对手写数字的识别。

注意：前5000个比后5000个要规整，这是因为前5000个数据来自于美国人口普查局的员工，而后5000个来自于大学生。

**（二） MNIST 数据集特征**

print(x\_train.shape, y\_test.shape)

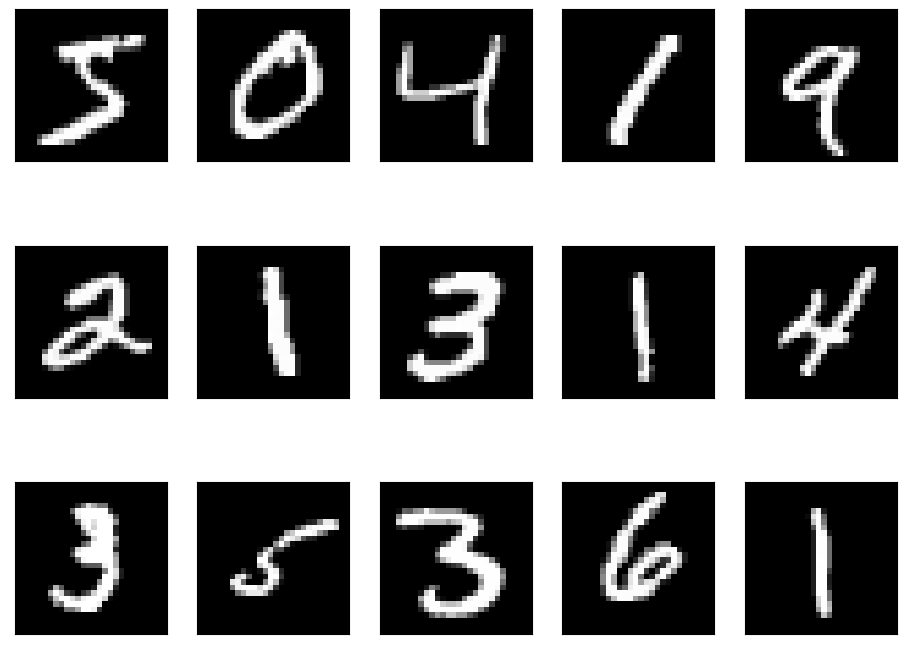


训练集train一共包含了 60000 张图像和标签，而测试集一共包含了 10000 张图像和标签。idx3表示3维，ubyte表示是以字节的形式进行存储的，t10k表示10000张测试图片（test10000）。每张图片是一个28\*28像素点的0 ~ 9的手写数字图片，图像像素值为0 ~ 255，越大该点越白。

**（三） MNIST 数据集内容**

*#画图 3行5列*n\_row=3  
n\_col=5  
plt.figure(figsize=(1.8\*n\_col,2.4\*n\_row))  
plt.subplots\_adjust(bottom=0,left=.01,right=.99,top=.90,hspace=.35)  
for i in range(n\_row\*n\_col):  
 plt.subplot(n\_row,n\_col,i+1)  
 plt.imshow(x\_train[i],cmap=plt.cm.gray)*#plt.cm.gray看灰度图* plt.xticks(())*#去掉x坐标* plt.yticks(())  
  
  
plt.show()

展示了前15个数据的内容



**（四）以灰度值查看数字图像**

*# 灰度函数返回数字*def plot\_digit(digit, dem = 28, font\_size = 12):  
 max\_ax = font\_size \* dem  
  
 fig = plt.figure(figsize=(13, 13))  
 plt.xlim([0, max\_ax])  
 plt.ylim([0, max\_ax])  
 plt.axis('off')  
 black = '#000000'  
  
 for idx in range(dem):  
 for jdx in range(dem):  
  
 t = plt.text(idx \* font\_size, max\_ax - jdx\*font\_size, digit[jdx][idx], fontsize = font\_size, color = black)  
 c = digit[jdx][idx] / 255.  
 t.set\_bbox(dict(facecolor=(c, c, c), alpha = 0.5, edgecolor = 'black'))  
  
 plt.show()

定义完成以灰度值返回数字图像函数，下面随机选取图像进行输出

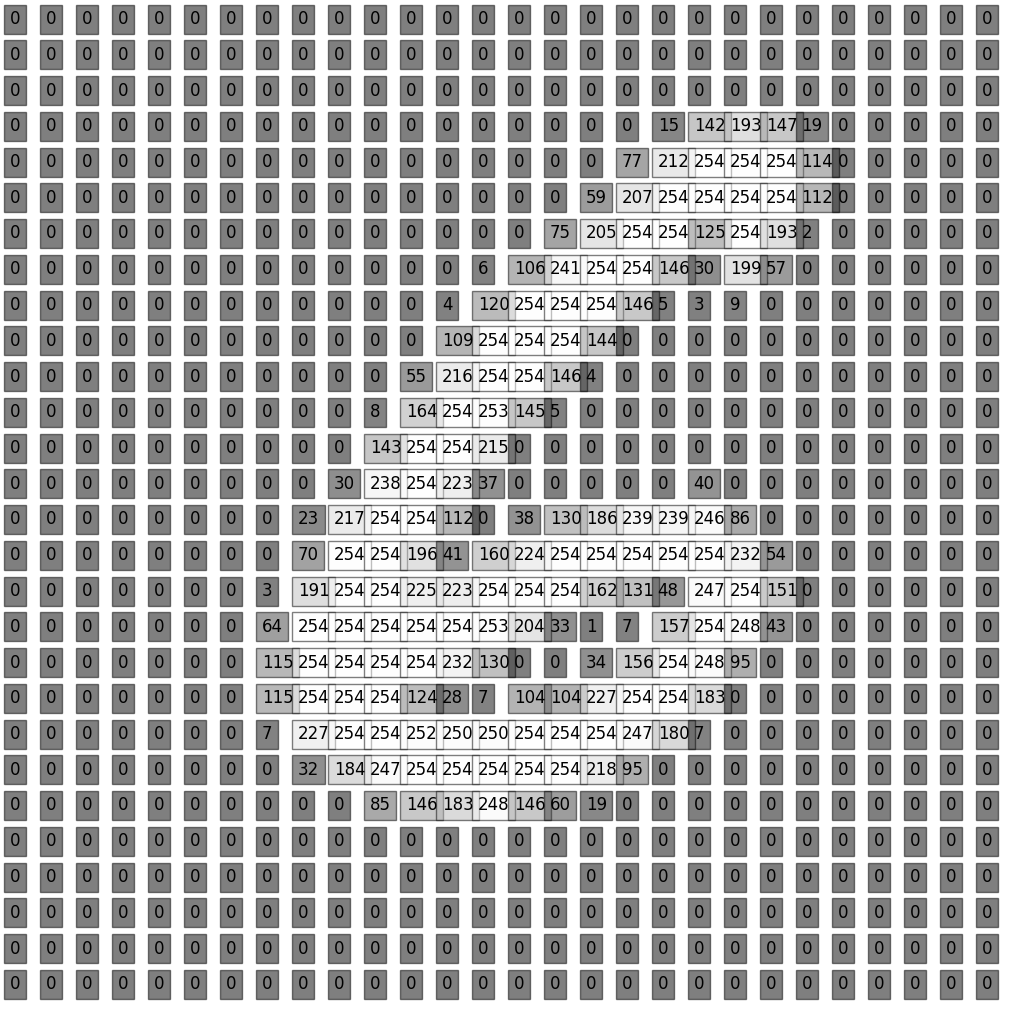
train\_images\_plot = x\_train.reshape(-1, 28, 28)  
  
rand\_number = random.randint(0, len(y\_train))  
print(y\_train[rand\_number])  
plot\_digit(train\_images\_plot[rand\_number])

random.randint产生随机整数。返回从低（包括）到高（不包括）的随机整数。 从“半开”区间 [low, high) 中指定 dtype 的“离散均匀”分布返回随机整数。

**标签为：**



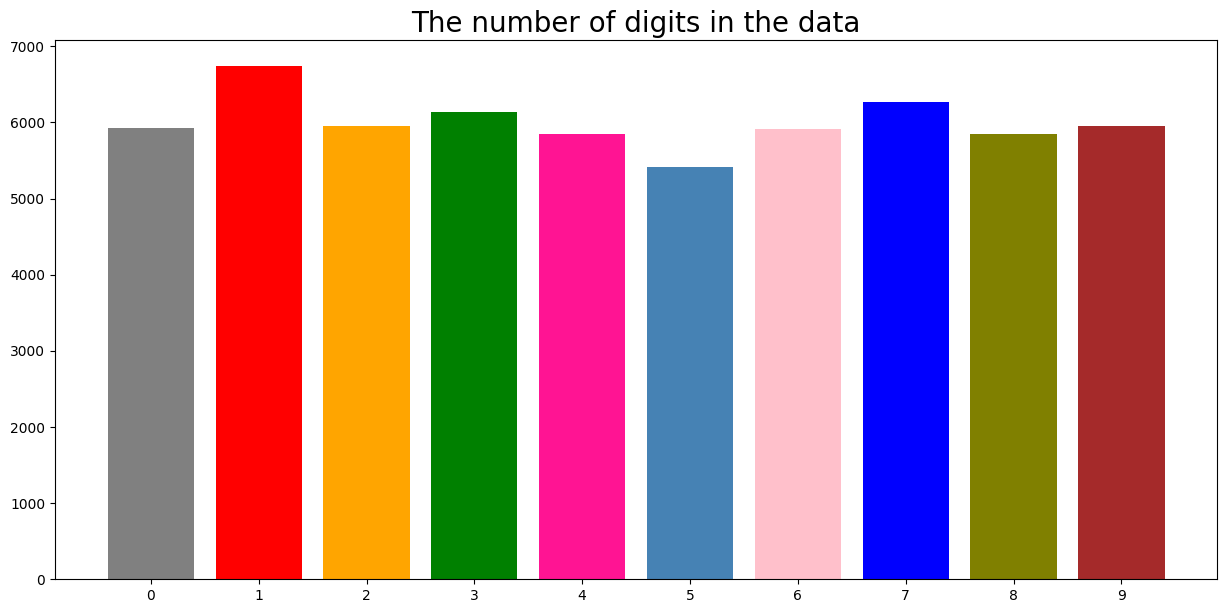
**输出图像为：**

****

**（五）查看不同标签数量**

digit\_range = np.arange(10)  
  
*###########*train\_labels=pd.Series(y\_train)  
  
val = train\_labels.value\_counts().index  
cnt = train\_labels.value\_counts().values  
mycolors = ['red', 'blue', 'green', 'orange', 'brown', 'grey', 'pink', 'olive', 'deeppink', 'steelblue']  
  
plt.figure(figsize = (15, 7))  
plt.title("The number of digits in the data", fontsize = 20)  
plt.xticks(range(10))  
plt.bar(val, cnt, color = mycolors);

在pandas中，value\_counts常用于数据表的计数及排序，它可以用来查看数据表中，指定列里有多少个不同的数据值，并计算每个不同值有在该列中的个数，同时还能根据需要进行排序。



**四、数据预处理**

**（一）数据格式转换**

x\_train = x\_train.reshape(-1,28,28,1)/255.0  
x\_test = x\_test.reshape(-1,28,28,1)/255.0

在 tensorflow 中，在做卷积的时候需要把数据变成 4 维的格式，这 4 个维度是(数据数量，图片高度，图片宽度，图片通道数)。

所以这里把数据 reshape 变成 4 维数据，黑白图片的通道数是 1，彩色图片通道数是 3。

**（二）独热编码**

独热编码是使用N位状态寄存器来对N个状态进行编码，每个状态都有它独立的寄存器位，并且在任意时候，其中只有一位有效。

y\_train = to\_categorical(y\_train,num\_classes=10)  
y\_test = to\_categorical(y\_test,num\_classes=10)

把训练集和测试集的标签转为独热编码

使用独热编码，将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点。将离散型特征使用One-Hot Encoding，会让特征之间的距离计算更加合理。

**五、CNN算法描述**

主要是不同layer的构建

**（一）卷积层**

卷积层的作用是提取一个局部区域的特征，不同的卷积核相当于不同的特征提取器．由于卷积网络主要应用在图像处理上，而图像为二维结构，因此为了更充分地利用图像的局部信息，通常将神经元组织为三维结构的神经层，其大小为 高度𝑀×宽度𝑁×深度𝐷，由𝐷 个𝑀 × 𝑁 大小的特征映射构成．

不失一般性，假设一个卷积层的结构如下：

（1） 输入特征映射组：𝒳 ∈ ℝ𝑀×𝑁×𝐷为三维张量（Tensor），其中每个切片（Slice）矩阵𝑿𝑑 ∈ ℝ𝑀×𝑁 为一个输入特征映射，1 ≤ 𝑑 ≤ 𝐷；

（2） 输出特征映射组：𝒴 ∈ ℝ𝑀′×𝑁′×𝑃 为三维张量，其中每个切片矩阵𝒀𝑝 ∈ ℝ𝑀′×𝑁′ 为一个输出特征映射，1 ≤ 𝑝 ≤ 𝑃；

（3） 卷积核：𝒲 ∈ ℝ𝑈×𝑉×𝑃×𝐷 为四维张量，其中每个切片矩阵𝑾𝑝,𝑑 ∈ ℝ𝑈×𝑉 为一个二维卷积核，1 ≤ 𝑝 ≤ 𝑃, 1 ≤ 𝑑 ≤ 𝐷．

为了计算输出特征映射 𝒀𝑝，用卷积核𝑾𝑝,1, 𝑾𝑝,2, ⋯ , 𝑾𝑝,𝐷 分别对输入特征映射 𝑿1, 𝑿2, ⋯ , 𝑿𝐷 进行卷积，然后将卷积结果相加，并加上一个标量偏置 𝑏𝑝得到卷积层的净输入𝒁𝑝。再经过非线性激活函数后得到输出特征映射𝒀𝑝．





其中𝑾𝑝 ∈ ℝ𝑈×𝑉×𝐷 为三维卷积核，𝑓(⋅)为非线性激活函数，一般用ReLU函数．

**（二）汇聚层**

汇聚层也叫子采样层，其作用是进行特征选择，降低特征数量，从而减少参数数量．

卷积层虽然可以显著减少网络中连接的数量，但特征映射组中的神经元个数并没有显著减少．如果后面接一个分类器，分类器的输入维数依然很高，很容易出现过拟合．为了解决这个问题，可以在卷积层之后加上一个汇聚层，从而降低特征维数，避免过拟合．

假设汇聚层的输入特征映射组为𝒳 ∈ ℝ𝑀×𝑁×𝐷，对于其中每一个特征映射

𝑿𝑑 ∈ ℝ𝑀×𝑁 , 1 ≤ 𝑑 ≤ 𝐷，将其划分为很多区域 𝑅𝑑𝑚,𝑛, 1 ≤ 𝑚 ≤ 𝑀′, 1 ≤ 𝑛 ≤ 𝑁′，这些区域可以重叠，也可以不重叠．汇聚是指对每个区域进行下采样得到一个值，作为这个区域的概括．

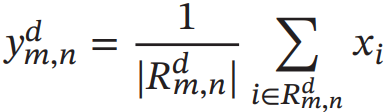
常用的汇聚函数有两种：

（1） 最大汇聚：对于一个区域𝑅𝑑𝑚,𝑛，选择这个区域内所有神经元的最大活性值作为这个区域的表示，即



其中𝑥𝑖 为区域𝑅𝑑𝑘 内每个神经元的活性值．

（2） 平均汇聚：一般是取区域内所有神经元活性值的平均值，即



对每一个输入特征映射𝑿𝑑 的𝑀′ × 𝑁′ 个区域进行子采样，得到汇聚层的输出特征映射𝒀𝑑 = {𝑦𝑑𝑚,𝑛}, 1 ≤ 𝑚 ≤ 𝑀′, 1 ≤ 𝑛 ≤ 𝑁′．

**六、CNN设计**

原始的手写数字的图片是一张 28×28 的图片，并且是黑白的，所以图片的通道数是1，输入数据是 28×28×1 的数据，如果是彩色图片，图片的通道数就为 3。

我构建的网络结构是一个 4 层的卷积神经网络。

**（一）第一层**

第 1 层为卷积层，使用 32 个 5×5 的卷积核对原始图片求卷积，步长为 1，Same Padding。因为是 Same Padding 并且步长为 1，所以卷积后的特征图大小跟原图片一样，可以得到 32 张 28×28 的特征图。池化的计算是在卷积层中进行的，使用 2×2，步长为 2 的池化窗口做池化计算，池化后得到 32 张 14×14 的特征图。特征图的长宽都变成了之前的 1/2。权值的数量为 5×5×32=800，偏置值数量为 32。

**（二）第二层**

第 2 层也是卷积层，使用 64 个 5×5 的卷积核对 32 张 14×14 的特征图求卷积，步长为1，Same Padding。因为是 Same Padding 并且步长为 1，所以卷积后的特征图大小跟原图片一样，可以得到 64 张 14×14 的特征图。

对多张特征图求卷积，相当于是同时对多张特征图进行特征提取。同一个特征图中权值是共享的，不同的特征图之间权值是不同的。对 32 张图像求卷积产生 1 个特征图，需要使用 32个不同的 5×5 的卷积核，那么就会有 5×5×32=800 个连接，800 个权值。

第 2 个卷积层卷积窗口大小 5×5，对 32 张图像求卷积产生 64 个特征图，参数个数是 5×5×32×64=51200 个权值加上 64 个偏置。

池化的计算是在卷积层中进行的，使用 2×2，步长为 2 的池化窗口做池化计算，池化后得到 64 张 7×7 的特征图。特征图的长宽都变成了之前的 1/2。

**（三）第三层**

第 3 层是全连接层，第 2 个池化层之后的 64×7×7 个神经元跟 1024 个神经元做全连接。

**（四）第四层**

第 4 层是输出层，输出 10 个预测值，对应 0-9 的 10 个数字。

**七、CNN实现**

定义顺序模型

model = Sequential()

**（一）第一层卷积**

第一个卷积层

model.add(Convolution2D(  
input\_shape = (28,28,1),  
filters = 32,  
kernel\_size = 5,  
strides = 1,  
padding = 'same',  
activation = 'relu'  
))

第一个池化层

model.add(MaxPooling2D(  
pool\_size = 2,  
strides = 2,  
padding = 'same',  
))

**（二）第二层卷积**

第二个卷积层

model.add(Convolution2D(64,5,strides=1,padding='same',activation='relu'))

第二个池化层

model.add(MaxPooling2D(2,2,'same'))

把第二个池化层的输出进行数据扁平化

model.add(Flatten())

**（三）全连接**

第一个全连接层

model.add(Dense(1024,activation = 'relu'))

Dropout

model.add(Dropout(0.5))

第二个全连接层

model.add(Dense(10,activation='softmax'))

**（四）优化器**

定义优化器

adam = Adam(lr=1e-4)

**（五）训练并保存模型**

定义优化器，loss function，训练过程中计算准确率

model.compile(optimizer=adam,loss='categorical\_crossentropy',metrics=['accuracy'])

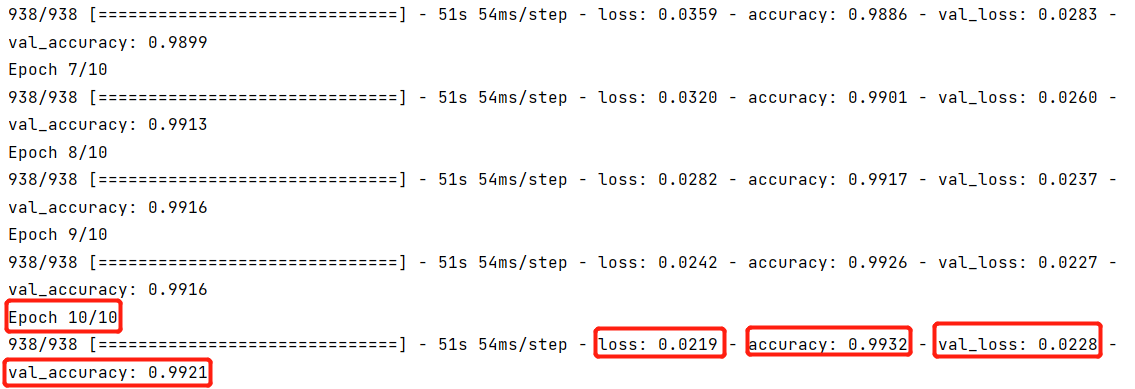
训练模型

model.fit(x\_train,y\_train,batch\_size=64,epochs=10,validation\_data=(x\_test, y\_test))

保存模型

model.save('mnist.h5')

运行过程：



**八、应用**

自己手写了一张数字，跟 MNIST 数据集中的数字类似

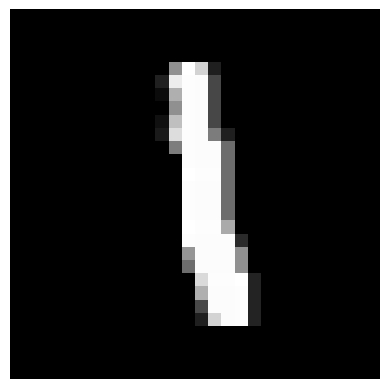


然后通过模型mnist.h5来完成数字图片的识别

**（一）MNIST示例**

获取MNIST一张照片,这里我们选择和我手写数字一样的**1**,用灰度图显示

plt.imshow(x\_train[6],cmap='gray')  
*# 不显示坐标*plt.axis('off')  
plt.show()



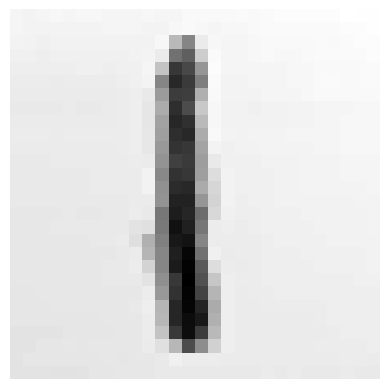
**（二）处理自己手写图片**

载入我自己写的数字图片

img=Image.open('1.jpg')  
*# 显示图片*plt.imshow(img)  
*# 不显示坐标*plt.axis('off')  
plt.show()

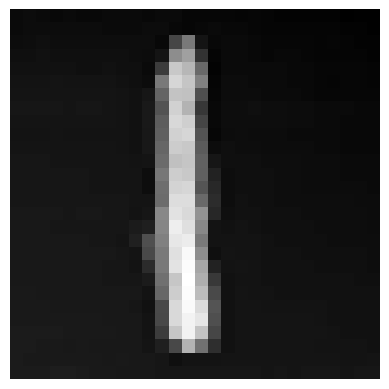


把图片大小变成 28×28，并且把它从 3D 的彩色图变为 1D 的灰度图  
image = np.array(img.resize((28,28)).convert('L'))  
*# 显示图片,用灰度图显示*plt.imshow(image,cmap='gray')  
*# 不显示坐标*plt.axis('off')  
plt.show()



观察发现我自己写的数字是白底黑字，MNIST 数据集的图片是黑底白字。所以需要先把图片从白底黑字变成黑底白字

image = (255-image)/255.0  
*# 显示图片，用灰度图显示*plt.imshow(image,cmap='gray')  
*# 不显示坐标*plt.axis('off')  
plt.show()



把数据处理变成 4 维数据

image = image.reshape((1,28,28,1))

**（三）开始训练**

载入训练好的模型

model = load\_model('mnist.h5')

对数据进行预测并得到它的类别

predict\_x=model.predict(image)  
classes\_x=np.argmax(predict\_x,axis=1)  
prediction = classes\_x  
print(prediction)

输出结果：



预测正确

**九、总结**

**（一）CNN**

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络。卷积神经网络具有表征学习能力，能够按其阶层结构对输入信息进行平移不变分类。

卷积神经网络仿造生物的视知觉机制构建，可以进行监督学习和非监督学习，其隐含层内的卷积核参数共享和层间连接的稀疏性使得卷积神经网络能够以较小的计算量对格点化特征，例如像素和音频进行学习、有稳定的效果且对数据没有额外的特征工程要求。

**（二）连接性**

卷积神经网络中卷积层间的连接被称为稀疏连接，即相比于前馈神经网络中的全连接，卷积层中的神经元仅与其相邻层的部分，而非全部神经元相连。

卷积神经网络的稀疏连接具有正则化的效果，提高了网络结构的稳定性和泛化能力，避免过度拟合，同时，稀疏连接减少了权重参数的总量，有利于神经网络的快速学习，和在计算时减少内存开销。

**（三）表征学习**

卷积神经网络中的卷积层和池化层能够响应输入特征的平移不变性，即能够识别位于空间不同位置的相近特征。能够提取平移不变特征是卷积神经网络在计算机视觉问题中得到应用的原因之一。