

# 西安科技大学本科生教学案例库建设项目

案例库名称： 《神经网络与深度学习》教学案例库

适用课程： 神经网络与深度学习

专业名称： 工科专业

学院名称： 计算机学院

项目负责人： 牟琦

2022 年 11 月 6 日

# 《神经网络与深度学习》混合式教学案例库

《神经网络与深度学习》混合式教学案例库包括“基础理论案例”、“专题技术案例”、“综合应用案例”三类案例，每个案例提供案例简介、原理与方法、程序说明及源码解析、程序运行指南、参考文献等文档，和完整的源代码和数据。

使用案例库展开教学，有利于将理论与实践应用相结合，引导学生从主动获取专业知识、发现问题、分析和解决问题，培养运用理论和方法解决实际问题的能力，从而提高新工科专业学生的人工智能实践能力，培养创新能力，充分发挥学生主观能动性，满足综合能力培养目标达成度要求，达到技能与知识、方法与过程的统一，对于切实提升工程教育人才培养质量具有广泛的应用前景价值和重要的现实意义。

## (1) 基础理论案例

基础理论案例主要用于课堂讲授、计划内实验和课后作业，要求学生根据要求完成任务并撰写实验报告。其内容设计与课堂讲授知识点进度同步，通过对同一任务不断迭代、层次深入的迭代和改进，帮助学生系统的学习相关理论知识，为进一步的综合应用打下扎实的基础。

表 1 基础理论案例

章节内容	案例名称	知识点
科学计算与数据可视化	A1.1 波士顿房价数据集可视化	NumPy 多维组数应用 访问 TensorFlow 内置数据集
	A1.2 鸢尾花数据集可视化	多维数组运算、切片等 使用 Pandas 访问外部数据集
	A1.3 手写数字数据集可视化	数字图像基础 PIL 应用
回归问题	A2.1 自建数据集房价预测-1	解析法实现一元线性回归
	A2.2 自建数据集房价预测-2	解析法实现多元线性回归
	A2.3 自建数据集房价预测-3	三维模型可视化
梯度下降法	A3.1 波士顿房价预测-1	梯度下降法实现一元线性回归
	A3.2 波士顿房价预测-2	TensorFlow 的自动求导机制
	A3.3 波士顿房价预测-3	数据分析与多元线性回归
分类问题	A4.1 自建数据集商品房分类-1	一元逻辑回归
	A4.2 鸢尾花二分类-1	多元逻辑回归 绘制分类图
	A4.3 鸢尾花三分类-2	多分类问题 Softmax 回归
人工神经网络	A4.4 鸢尾花三分类-1	单层神经网络
	A4.5 鸢尾花三分类-2	多层神经网络

卷积神经网络	A5.1 手写数字识别-1	keras.Sequential 模型
	A5.2 手写数字识别-2	卷积神经网络
	A6.1 Cifar10 图像分类	卷积神经网络的优化

### (2) 专题技术案例

专题技术案例专题选取深度学习的典型专题和经典论文,要求学生了解并复现专题领域经典论文和典型算法,并对比实验结果、分析原因、撰写学习总结报告;可用于专题讨论课、期末考核或课程设计。

**表 2 专题技术案例**

案例序号	案例名称	专题
B1	基于 MSR-net 的图像增强网络	图像增强
B2	基于 KinD 算法的图像增强网络	图像增强
B3	基于全局单应性的自由视角图像拼接网络	图像拼接
B4	基于无监督深度学习的图像拼接	图像拼接
B5	基于全卷积孪生网络的目标跟踪	目标跟踪
B6	基于 VGG19 的图像风格迁移	图像风格迁移

### (3) 综合应用案例

综合应用案例以某一具体场景为基础,要求学生自主选择 and 建立模型,按要求完成任务,并分析结果,撰写项目总结报告;可用于专题讨论课、期末考核或课程设计。

**表 3 综合应用案例**

案例序号	案例名称	案例内容
C1	基于 PCA 和全连接神经网络的乳腺癌预测	案例简介 原理与方法 程序说明及源码解析 程序运行指南 数据和程序源代码
C2	基于 CNN 的影评文本分类	
C3	基于卷积神经网络的验证码识别	
C4	基于 TensorFlow.Lite 的手写数字识别应用	
C5	基于 TensorFlow.js 的手写数字识别应用	
C6	猫狗大战——数据增强在深度学习中的应用	
C7	泰坦尼克号旅客生存概率预测	
C8	美国爱荷华州艾姆斯房价预测	
C9	基于卷积神经网络的表情识别	
C10	基于深度学习的车牌识别	
C11	基于 CNN 的手写汉字识别	
C12	基于 TensorFlow.js 的人体姿态估计	
C13	基于 YOLOv5 的农田杂草识别	

**基础理论案例**百度网盘链接:

**链接:** <https://pan.baidu.com/s/1mPYGAYRZlgpNPZ-SHQhovw?pwd=1234>

**专题技术案例**百度网盘链接:

**链接:** <https://pan.baidu.com/s/146LSG3xfSVNXHuAgwsOqeA?pwd=1234>

**综合应用案例**百度网盘链接:

**链接:** <https://pan.baidu.com/s/1nsaFodbFwbe7WrwoFC8aPQ?pwd=1234>

# A1.1 波士顿房价数据集可视化：访问 TensorFlow 内置数据集应用

本案例使用 TensorFlow 内置的波士顿房价数据集，使用 Matplotlib 实现数据集的可视化。

## 1 学习目标

- ① 了解波士顿房价数据集；
- ② 了解 Keras 库；
- ③ 掌握访问和加载波士顿房价数据集的基本方法；
- ④ 掌握 Numpy 多维数组应用；
- ⑤ 掌握使用 Matplotlib 的绘图方法；
- ⑥ 能够使用 Matplotlib 实现对波士顿房价数据集可视化。

## 2 基础知识

### 2.1 Keras 库

Keras 是一个高层的神经网络和深度学习库，可以快速搭建神经网络模型，非常易于调试和扩展，是 TensorFlow 的官方 API。Keras 中内置了一些常用的公共数据集，如波士顿房价数据集、手写数字图片集等，可以通过 `keras.datasets` 模块加载和访问。

### 2.2 Matplotlib 绘图库

Matplotlib 是 Python 中最常用的可视化工具之一，可以非常方便地创建海量类型地 2D 图表和一些基本的 3D 图表，可根据数据集（`DataFrame`，`Series`）自行定义 `x`，`y` 轴，绘制图形（线形图，柱状图，直方图，密度图，散布图等等），能够解决大部分的需要。Matplotlib 中最基础的模块是 `pyplot`。

### 2.3 散点图

散点图是数据点在直角坐标系中的分布图。`pyplot` 中使用 `scatter()` 函数绘制散点图。

## 3 波士顿房价数据集

波士顿房价数据集取自卡内基梅隆大学维护的 StatLib 库，1978 年开始统计，涵盖了麻省波士顿的 506 个不同郊区的房屋数据。

一共含有 506 条数据，404 条训练数据集，102 条测试数据集。每条数据 14 个字段，包含 13 个属性，和一个房价的平均值。如下所示：

下载数据集：

由于波士顿房价数据集是 Keras 中的内置数据集，可以 `keras.datasets` 模块访问数据集，使用 `load.data()` 方法加载数据集。

```
boston_housing = tf.keras.datasets.boston_housing
(train_x, train_y), (test_x, test_y) = boston_housing.load_data(test_split=0)
```

## 4 案例要求

下载波士顿数房价数据集，并绘制数据集中各个属性与房价之间的散点图，实现数据集可视化；要求用户选择属性，如图 A1.1.2 所示，根据用户的选择，输出对应属性的散点图，如图 A1.1.3 所示。

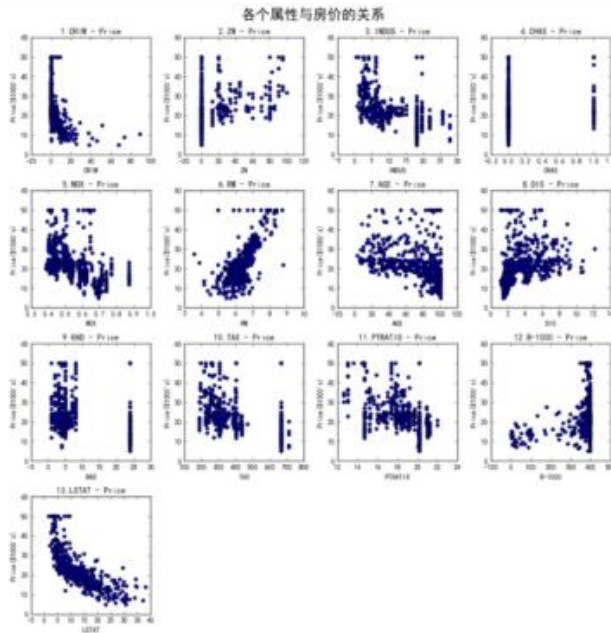


图 A1.1.1 波士顿房价数据集可视化图

请用户输入属性:

请选择属性:

- 1 -- CRIM
- 2 -- ZN
- 3 -- INDUS
- 4 -- CHAS
- 5 -- NOX
- 6 -- RM
- 7 -- AGE
- 8 -- DIS
- 9 -- RAD
- 10 -- TAX
- 11 -- PTRATIO
- 12 -- B
- 13 -- LSTAT

图 A1.1.2 属性选择图

运行结果:

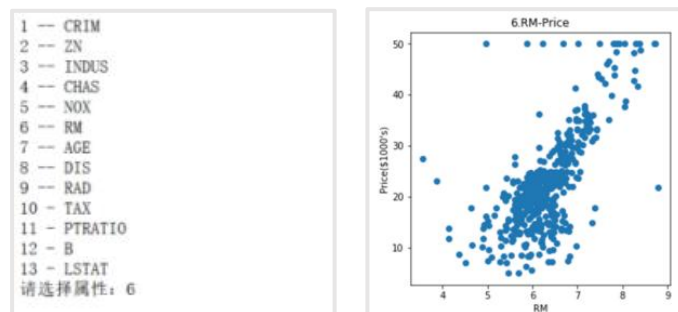


图 A1.1.3 运行结果图

(1) 下载波士顿数据集，读取全部 506 条数据，放在 NumPy 数组  $x$ 、 $y$  中 ( $x$ : 属性,  $y$ :

标记)；

(2) 使用全部 506 条数据，编写代码，实现波士顿房价数据集可视化，如图 A1.1.1 所示；

(3) 分析结果，观察波士顿房价可视化结果，分析各个属性对房价的影响。如果仅选择 1-2 个属性，能够准确的预测出波士顿房价吗？通过以上结果，可以得到什么结论，或对你有什么启发。

## A1.2 鸢尾花数据集可视化：Pandas 访问外部数据集

本案例使用 Pandas 加载鸢尾花数据集，使用 Matplotlib 实现对鸢尾花数据集可视化。

### 1 学习目标

- ① 了解鸢尾花数据集；
- ② 掌握 Pandas 访问 csv 文件的基本方法；
- ③ 掌握多维数组应用，切片等；
- ④ 掌握使用 Matplotlib 创建绘图对象、划分子图等的方法；
- ⑤ 掌握绘制散点图的实现方法；
- ⑥ 能够使用 Pandas 加载鸢尾花数据集，使用 Matplotlib 实现对鸢尾花数据集可视化。

### 2 基础知识

#### 2.1 Matplotlib 绘图库

Matplotlib 是 Python 中最常用的可视化工具之一，可以非常方便地创建海量类型地 2D 图表和一些基本的 3D 图表，可根据数据集（DataFrame，Series）自行定义 x，y 轴，绘制图形（线形图，柱状图，直方图，密度图，散布图等等），能够解决大部分的需要。Matplotlib 中最基础的模块是 pyplot。

#### 2.2 散点图

散点图是数据点在直角坐标系中的分布图。pyplot 中使用 scatter() 函数绘制散点图。

#### 2.3 Pandas 库

Pandas 是基于 NumPy 的一个开源 Python 库，它被广泛用于快速分析数据，以及数据清洗和准备等工作。Pandas 能很好地处理来自各种不同来源的数据，比如 Excel 表格、CSV 文件、SQL 数据库，甚至还能处理存储在网页上的数据，常常与 Numpy、matplotlib 一起使用。

### 3 Iris 数据集

Iris 数据集是常用的分类实验数据集，由 Fisher, 1936 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（Setosa, Versicolour, Virginica）三个种类中的哪一类。

下载数据集：

通过 keras.utils 模块，使用 get\_file() 函数进行下载，read\_csv() 读取数据。

```
TRAIN_URL = "http://download.tensorflow.org/data/iris_training.csv"
train_path=tf.keras.utils.get_file(TRAIN_URL.split('/')[-1],TRAIN_URL)
TEST_URL = "http://download.tensorflow.org/data/iris_test.csv"
test_path=tf.keras.utils.get_file(TEST_URL.split('/')[-1],TEST_URL)
```

### 4 案例要求

使用鸢尾花数据集，绘制如下图形（如图 A1.2.1），其中对角线为属性的直方图。



Anderson's Iris Data Set  
(Blue->Setosa | Red->Versicolor | Green->Virginica)

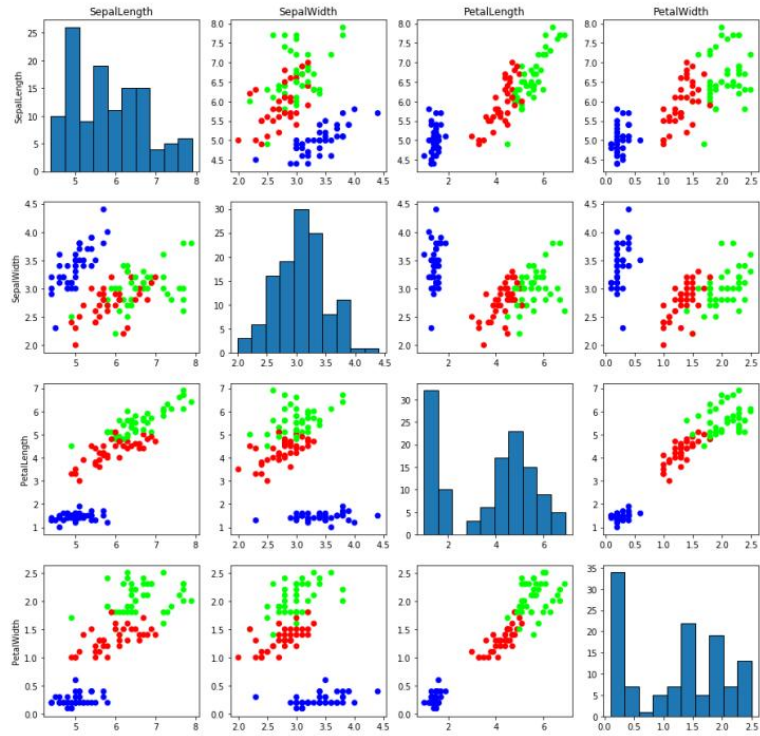


图 A1.2.1 鸢尾花数据集

- (1) 下载鸢尾花数据集。
- (2) 编写代码，将鸢尾花数据可视化。
- (3) 分析结果：观察鸢尾花数据集可视化结果，如果要将山鸢尾和其他两种鸢尾分开，至少需要几种属性？如果要将三种鸢尾花都分开，至少需要几种属性；通过以上结果，可以得到什么结论，或对你有什么启发。

## A1.3 手写数字数据集可视化：数字图像基础应用

本案例根据数字图形基础，使用 Matplotlib 实现 MNIST 数据集样本可视化。

### 1 学习目标

- ① 了解 MNIST 数据集；
- ② 了解 Pillow 图像处理库；
- ③ 理解位图、图像格式、色彩模式等数字图像基础知识；
- ④ 掌握 Matplotlib 中显示图片的方法；
- ⑤ 能够使用 Matplotlib 实现输出 MNIST 数据集样本的图片形式。

### 2 基础知识

#### 2.1 Matplotlib 绘图库

Matplotlib 是 Python 中最常用的可视化工具之一，可以非常方便地创建海量类型地 2D 图表和一些基本的 3D 图表，可根据数据集（DataFrame，Series）自行定义 x，y 轴，绘制图形（线形图，柱状图，直方图，密度图，散布图等等），能够解决大部分的需要。Matplotlib 中最基础的模块是 pyplot。

#### 2.2 Pillow

PIL(Python Image Library)是 python 的第三方图像处理库，因其强大的功能与众多的使用人数，但由于 PIL 仅支持到 Python 2.7，于是一群志愿者在 PIL 的基础上创建了兼容的版本，名字叫 Pillow，支持最新 Python 3.x，又加入了许多新特性。

Pillow 作为 python 的第三方图像处理库，提供了广泛的文件格式支持，强大的图像处理能力，主要包括图像储存、图像显示、格式转换以及基本的图像处理操作等。

### 3 MNIST 数据集

MNIST 数据集由 Yann LeCun 搜集，是一个大型的手写体数字数据库，通常用于训练各种图像处理系统，也被广泛用于机器学习领域的训练和测试。MNIST 数字识别数据集数据量不会太多，而且是单色的图像，较简单，适合深度学习初学者练习建立模型、训练、预测。

MNIST 数据集共有训练数据 60000 项、测试数据 10000 项。数据集是有 250 人手写数字组成，一半是高中生，一半是美国人口普查局。每张图像的大小为 28\*28（像素），每张图像都为灰度图像，位深度为 8（灰度图像是 0-255）。但并不是有以图片形式存储的，而是存储在一个 28\*28 的二维数组中，数组中的每个元素对应于图片中的一个像素。MNIST 数据集已经被集成在 Keras 中了，可以直接使用 Keras 中的 datasets 模块访问。

下载数据集：

通过 keras 模块中的 datasets 模块访问：

```
mnist = tf.keras.datasets.mnist
(train_x, train_y), (test_x, test_y) = mnist.load_data()
```

### 4 案例要求

按下列要求完成程序，随机显示 MNIST 数据集中的样本，效果如下图 A1.3.1 所示。



图 A1.3.1 MNIST 测试集样本

- (1) 下载 MNIST 数据集。
- (2) 编写代码，读取训练集和测试集数据，放在 NumPy 数组 `train_x`、`train_y`、`test_x`、`test_y` 中；（`train_x`：训练集图像，`train_y`：训练集标签，`test_x`：测试集图像，`test_y`：测试集标签）
- (3) 随机从所有测试集数据中显示 16 幅数字图像；
- (4) 16 幅图像按照 4×4 方式排列在一张画布中，每幅图像的子标题为该图像的标签值，字体大小为 14，全局标题为“MNIST 测试集样本”，字体大小为 20，颜色为红色；
- (5) 分析结果：通过以上结果，可以得到什么结论，或对你有什么启发。

## A2.1 自建数据集房价预测：解析解实现一元线性回归

本案例通过求解模型参数解析解，使用一元线性回归模型实现对自建数据集房价预测。

### 1 学习目标

- ① 了解机器学习基本概念；
- ② 理解线性回归模型的基本原理；
- ③ 掌握解析解实现一元线性回归模型的方法；
- ④ 掌握平方损失函数的计算方法；
- ⑤ 掌握使用 Python、NumPy、TensorFlow 求模型参数解析解的方法；
- ⑥ 能够使用一元线性回归实现对自建数据集房价预测。

### 2 基础知识

#### 2.1 线性回归模型

回归主要用于预测数值型数据，根据观测到的数据，设计一种模型，描述数据之间蕴含的关系，通过给定的数值点，拟合出最优的曲线（回归曲线）。

线性回归模型是利用线性拟合的方式探寻数据背后的规律。再利用回归曲线进行一些简单的预测分析或因果关系分析。在线性回归中，根据特征变量（也称自变量）来预测反应变量（也称因变量）。根据特征变量的个数可将线性回归模型分为一元线性回归和多元线性回归。

#### 2.2 一元线性回归模型

一元线性回归模型又称为简单线性回归模型，形式可以表示为： $y=wx+b$

其中， $x$  称为模型变量； $w$ 、 $b$  称为模型参数， $w$  为权重， $b$  为偏置值。

如图 A2.1.1 所示，其中  $y^{(i)}$  为实际值， $\hat{y}^{(i)}$  为预测值。

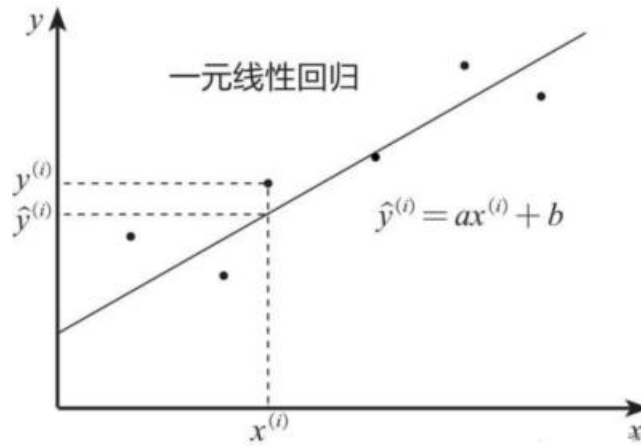


图 A2.1.1 一元线性回归

一元线性回归的目的就是拟合出一条线来使得预测值和实际值尽可能接近，如果大部分点都落在拟合出来的线上，则该线性回归模型拟合得较好。

#### 2.3 平方损失函数

损失函数用来评价模型的预测值和真实值不一致的程度，它是一个非负实值函数。损失函数越好，通常模型的性能越好，不同的模型用的损失函数一般也不一样。

在一元线性回归模型中，使用平方损失函数计算模型预测值与真实值的不一致程度，平

方损失函数不仅计算方便，并且所找到的直线也是总体最接近这些点的直线。公式如下：

$$Loss = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{2} \sum_{i=1}^n (y_i - (wx_i + b))^2$$

其中  $y_i$  为实际值， $\hat{y}_i$  为预测值。

除了平方损失函数外，还有经常使用均方误差，来作为衡量误差的指标：

$$Loss = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{2n} \sum_{i=1}^n (y_i - (wx_i + b))^2$$

## 2.4 解析解

解析解，是指通过严格的公式推导和计算所求得解，是一个封闭形式的函数，给出任意的自变量，就可以通过严格的公式求出准确的因变量，因此解析解也称为封闭解或闭式解。

通过样本数据，确定模型参数的解析解，确定最佳拟合曲线。经过严格的公式推导和计算，得到的解析解  $w$ 、 $b$  值，就是损失函数达到最小值的模型参数。公式如下：

$$w = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$b = \bar{y} - w\bar{x}$$

## 3 样本数据

使用“商品房销售记录”中的数据作为样本数据，如表 A2.1.1：

表 A2.1.1 商品房销售记录表

序号	面积 (平方米)	销售价格 (万元)	序号	面积 (平方米)	销售价格 (万元)
1	137.97	145.00	9	106.69	62.00
2	104.50	110.00	10	138.05	133.00
3	100.00	93.00	11	53.75	51.00
4	124.32	116.00	12	46.91	45.00
5	79.20	65.32	13	68.00	78.50
6	99.00	104.00	14	63.02	69.65
7	124.00	118.00	15	81.26	75.69
8	114.00	91.00	16	86.21	95.30

## 4 案例要求

使用“商品房销售记录表”作为样本数据，训练一元线性回归模型，根据商品房面积预测房价。

提示用户输入商品房面积，并进行输入校验。合理的输入为 20~500 之间的实数。如果输入正确，根据模型估计房价，并输出显示；如果输入数据类型错误，或者输入数据范围不合理，根据错误类型提示，并等待用户重新输入，输错 3 次，则程序结束。

要求：

- (1) 加载样本数据；
- (2) 尝试分别使用 Python、NumPy、TensorFlow 编写代码，实现程序功能；
- (3) 分析结果，记录实验过程和结果。通过以上结果，可以得到什么结论，或对你有什么启发。

## A2.2 自建数据集房价预测：解析解实现多元线性回归

本案例通过模型参数解析解，使用多元线性回归模型实现对自建数据集房价预测。

### 1 学习目标

- ① 了解多元线性回归模型的几种形式；
- ② 理解线性回归模型的基本原理；
- ③ 掌握求解多元线性回归模型参数解析解的方法；
- ④ 掌握多元模型平方损失函数的计算方法；
- ⑤ 能够使用多元线性回归实现对自建数据集房价预测。

### 2 基础知识

#### 2.1 多元线性回归模型

多元线性回归是指回归分析中包括两个或两个以上的自变量，因变量与自变量是线性关系。形式可以表示为：

$$\hat{y} = w_1x^1 + w_2x^2 + \dots + w_mx^m + b,$$

其中， $x^1, x^2, \dots, x^m$  表示样本的属性， $m$  是属性个数； $w_1, w_2, \dots, w_m$ ，是为不同属性赋予的不同权值。为便于表示为向量形式，将令  $b = w_0, x^0 = 1$ ；多元线性回归模型向量形式可以表示为：

$$\hat{y} = w_1x^1 + w_2x^2 + \dots + w_mx^m + b = W^T X$$

其中  $W = (w_0, w_1, \dots, w_m)^T$ ； $X = (x^0, x^1, \dots, x^m)^T$ 。

矩阵形式为： $XW = \hat{Y}$ 。

#### 2.2 平方损失函数

损失函数用来评价模型的预测值和真实值不一样的程度，它是一个非负实值函数。损失函数越好，通常模型的性能越好，不同的模型用的损失函数一般也不一样。

在多元线性回归模型中，使用平方损失函数计算模型预测值与真实值的不一致程度，平方损失函数不仅计算方便，并且所找到的直线也是总体最接近这些点的直线。多元线性回归的损失函数公式如下：

$$Loss = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - W^T X_i)^2$$

其中  $y_i$  为实际值， $\hat{y}_i$  为预测值， $i=1 \sim n$  表示有  $n$  个样本。

将损失函数表述为向量形式：

$$Loss = (Y - XW)^T (Y - XW)$$

#### 2.3 解析解

解析解，是指通过严格的公式推导和计算所求得解，是一个封闭形式的函数，给出任意的自变量，就可以通过严格的公式求出准确的因变量，因此解析解也称为封闭解或闭式解。

通过样本数据，确定模型参数的解析解，确定最佳拟合曲线。经过严格的公式推导和计算，得到权值  $W$  的解析解，就是是损失函数达到最小值的模型参数。 $W$  解析解公式如下：

$$W = (X^T X)^{-1} X^T Y$$

### 3 样本数据

使用下表“商品房销售记录”中的数据作为样本数据：

表 A2.2.1 商品房销售记录表

序号	面积 (平方米)	房间数	销售价格 (万元)	序号	面积 (平方米)	房间数	销售价格 (万元)
1	137.97	3	145.00	9	106.69	2	62.00
2	104.50	2	110.00	10	138.05	3	133.00
3	100.00	2	93.00	11	53.75	1	51.00
4	124.32	3	116.00	12	46.91	1	45.00
5	79.20	1	65.32	13	68.00	1	78.50
6	99.00	2	104.00	14	63.02	1	69.65
7	124.00	3	118.00	15	81.26	2	75.69
8	114.00	2	91.00	16	86.21	2	95.30

### 4 案例要求

使用“商品房销售记录表”作为样本数据，使用房屋面积和房间数，训练多元线性回归模型，实现一个房价预测系统。

提示用户输入商品房面积和房间数，并进行输入校验。如果输入正确，根据模型预测房价。

合理的输入如下：

面积：20-500 之间的实数

房间数：1-10 之间的整数

如果输入数据类型错误，或者输入数据范围不合理，根据错误类型给出提示，并等待用户重新输入，输错 3 次，则程序结束。

要求：

(1) 加载样本数据；

(2) 编写代码，实现程序功能；

(3) 分析结果，记录实验过程和结果。通过以上结果，可以得到什么结论，或对你有什么启发。在题目基本要求的基础上，你对每个题目做了哪些扩展和提升？或者你觉得在编程实现过程中，还有哪些地方可以进行优化？

提示：

① TensorFlow 中矩阵求逆函数 `tf.linalg.inv()`；

② 输入时，请注意房间面积对应房间数的合理性。



## A2.3 自建数据集房价预测：三维模型可视化的应用

本案例使用 mplot3d 工具集实现自建数据集房价预测可视化。

### 1 学习目标

- ① 了解 mplot3d 工具集；
- ② 掌握绘制三维图形的方法；
- ③ 能够使用 mplot3d 工具集实现自建数据集房价预测可视化。

### 2 基础知识

#### 2.1 多元线性回归模型

mplot3d 是 matplotlib 中专门绘制 3D 图表的工具包，它主要包含一个继承自 Axes 的子类 Axes3D，使用 Axes3D 类可以构建一个三维坐标系的绘图区域。

### 3 样本数据

使用下表“商品房销售记录”中的数据作为样本数据：

表 A2.3.1 商品房销售记录表

序号	面积 (平方米)	房间数	销售价格 (万元)	序号	面积 (平方米)	房间数	销售价格 (万元)
1	137.97	3	145.00	9	106.69	2	62.00
2	104.50	2	110.00	10	138.05	3	133.00
3	100.00	2	93.00	11	53.75	1	51.00
4	124.32	3	116.00	12	46.91	1	45.00
5	79.20	1	65.32	13	68.00	1	78.50
6	99.00	2	104.00	14	63.02	1	69.65
7	124.00	3	118.00	15	81.26	2	75.69
8	114.00	2	91.00	16	86.21	2	95.30

使用 A2.2 中得到的模型参数进行绘图：

$W = [11.96729030, 0.53488599, 14.33150378]$

### 4 案例要求

使用“商品房销售记录表”作为样本数据，根据 A2.2 中模型训练得到的模型参数，创建 3D 绘图对象来绘制空间点集。x 轴表示房屋面积，y 轴表示房间数，z 轴表示样本的销售价格；

要求：

- (1) 输入样本数据；
- (2) 编写代码，绘制散点图、平面图、线框图，实现房价预测模型可视化。
- (3) 分析结果：使用 view\_init() 函数改变视角，从不同角度绘图，观察图像。分析实验结果，可以得到什么结论，或者对你有什么启发。

# A3.1 波士顿房价预测：梯度下降法实现一元线性回归

本案例使用梯度下降法求解一元线性回归，实现对波士顿房价预测。

## 1 学习目标

- ① 理解梯度下降法的基本原理；
- ② 掌握梯度下降法更新模型参数的方法；
- ③ 掌握梯度下降法求解线性回归的方法；
- ④ 掌握求解模型参数数值解的方法；
- ⑤ 能够使用梯度下降法求解一元线性回归问题，实现对波士顿房价预测。

## 2 基础知识

### 2.1 数值解

通过某种近似计算得到的解，能够在给定的精度下满足方程。

### 2.2 梯度下降法

梯度下降法（英语：**Gradient descent**）是一个一阶最优化算法。要使用梯度下降法找到一个函数的局部极小值，必须向函数上当前点对应梯度（或者是近似梯度）的反方向的规定步长距离点进行迭代搜索。

梯度下降的基本过程就和下山的场景很类似。首先，我们有一个可微分的函数。这个函数就代表着一座山。我们的目标就是找到这个函数的最小值，也就是山底。最快的下山的方式就是找到当前位置最陡峭的方向，然后沿着此方向向下走，对应到函数中，就是找到给定点的梯度，然后朝着梯度相反的方向，就能让函数值下降的最快。重复利用这个方法，反复求取梯度，最后就能到达局部的最小值。

对于机器学习算法，只要能够把损失函数描述成凸函数，那么就一定可以采用梯度下降法，以最快速度更新权值向量  $w$ ，找到使损失函数达到最小值点的位置。能够在每次迭代时自动调节步长，确定更新方向。

一元凸函数迭代算法：

$$x^{(k+1)} = x^{(k)} - \eta \frac{df(x)}{dx}$$

二元凸函数迭代算法：

$$\begin{aligned}x^{(k+1)} &= x^{(k)} - \eta \frac{\partial f(x, y)}{\partial x} \\y^{(k+1)} &= y^{(k)} - \eta \frac{\partial f(x, y)}{\partial y}\end{aligned}$$

### 2.3 损失函数

平方损失函数是一个二元凸函数，可以根据梯度下降法，得到他的权值更新算法，将偏导数带入，得到最终的迭代公式。

$$Loss = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{2} \sum_{i=1}^n (y_i - (wx_i + b))^2$$

## 2.4 模型参数更新算法

使用梯度下降法求解一元线性回归，使用模型参数更新算法：

$$w^{(k+1)} = w^{(k)} - \eta \frac{\partial Loss}{\partial w}$$
$$b^{(k+1)} = b^{(k)} - \eta \frac{\partial Loss}{\partial b}$$

使用均方差损失函数时的模型更新算法：

$$\frac{\partial Loss}{\partial w} = \frac{1}{n} \sum_{i=1}^n x_i (wx_i + b - y_i)$$
$$\frac{\partial Loss}{\partial b} = \frac{1}{n} \sum_{i=1}^n (wx_i + b - y_i)$$

## 3 波士顿房价数据集

波士顿房价数据集取自卡内基梅隆大学维护的 StatLib 库，1978 年开始统计，涵盖了麻省波士顿的 506 个不同郊区的房屋数据。

一共含有 506 条数据，404 条训练数据集，102 条测试数据集。每条数据 14 个字段，包含 13 个属性，和一个房价的平均值。

下载数据集：

由于波士顿房价数据集是 Keras 中的内置数据集，可以 `keras.datasets` 模块访问数据集，使用 `load_data()` 方法加载数据集。

```
boston_housing = tf.keras.datasets.boston_housing
(train_x, train_y), (test_x, test_y) = boston_housing.load_data(test_split=0)
```

## 4 案例要求

使用波士顿房价数据集中的“低收入人口比例”属性，使用梯度下降法更新模型参数，训练一元线性回归模型，并测试其性能，以可视化的形式展现训练测试的过程。

- (1) 编写代码，建立、训练并测试模型。并以可视化的形式展现训练和测试的过程及结果；
- (2) 尝试调试学习率、迭代次数等超参数，使模型达到最优的性能，请记录超参数的调试过程及结果，并简要分析和总结；
- (3) 尝试使用波士顿房价数据集中的其他属性，训练模型；
- (4) 分析结果：比较选择不同属性时，两者的学习率、迭代次数等超参数、在训练集和测试集上的均方差损失、以及模型训练时间，以表格或其他合适的图表形式展示？说明选择某种属性的依据；通过以上结果，可以得到什么结论，或对你有什么启发。

## A3.2 波士顿房价预测：TensorFlow 自动求导机制应用

本案例使用 TensorFlow 实现梯度下降法，求解二元线性回归，实现对波士顿房价预测。

### 1 学习目标

- ① 了解 Variable 可训练变量；
- ② 掌握使用 TensorFlow 自动求导机制实现梯度下降法；
- ③ 掌握梯度下降法求解线性回归的方法；
- ④ 掌握数据处理的几种归一化方法；
- ⑤ 能够使用 TensorFlow 的可训练变量和自动求导机制实现梯度下降法，求解二元线性回归问题，实现对波士顿房价预测。

### 2 基础知识

#### 2.1 Variable 可训练变量

在学习 TensorFlow 的自动求导之前，先要知道什么是 Variable 对象。Variable 对象其实是对 tensor 对象的进一步封装，它在模型训练过程中自动记录梯度信息，并且由算法自动调整和优化它的取值。它是一个可以被训练的变量，在机器学习中作为模型参数，可以通过 assign()、assign\_add()、assign\_sub() 等函数对它进行赋值。

#### 2.2 TensorFlow 自动求导机制

TensorFlow 提供了一个专门用来求导的类 GradientTape，可以形象的理解为记录梯度数据的磁带。通过他，可以实现对变量的自动求导和监视。GradientTape 类实现了上下文管理器，它能够监视 with 语句块中所有的变量和计算过程，并把它们自动记录在梯度带中。以下是 GradientTape 类的构造函数。

```
With GradientTape() as tape:  
    函数表达式  
grad =tape.gradient(函数, 自变量)
```

#### 2.3 损失函数

平方损失函数是一个凸函数，可以根据梯度下降法，得到他的权值更新算法，将偏导数带入，得到最终的迭代公式。

平方损失函数的公式：

$$Loss = \frac{1}{2}(Y - \hat{Y})^2 = \frac{1}{2}(Y - XW)^2$$

其中， $W = (w_0, w_1, \dots, w_m)^T$ ； $X = (x^0, x^1, \dots, x^m)^T$ 。

#### 2.4 模型参数更新算法

使用梯度下降法求解二元线性回归，使用模型参数更新算法：

$$\frac{\partial Loss}{\partial W} = X^T(XW - Y)$$

$$W^{(k+1)} = W^{(k)} - \eta \frac{\partial Loss(W)}{\partial W}$$

$$W^{(k+1)} = W^{(k)} - \eta X^T(XW - Y)$$

其中，梯度信息使用 TensorFlow 自动求导机制对损失函数求导。

## 2.5 数据归一化/标准化

数据归一化是指将数据的值限制在一定范围之内。使所有属性处于同一个范围、同一数量级下。对数据进行归一化处理，可以使模型更快收敛到最优解，提高学习器的精度。归一化有线性归一化、标准差归一化和非线性映射归一化。

## 3 波士顿房价数据集

波士顿房价数据集取自卡内基梅隆大学维护的 StatLib 库，1978 年开始统计，涵盖了麻省波士顿的 506 个不同郊区的房屋数据。

一共含有 506 条数据，404 条训练数据集，102 条测试数据集。每条数据 14 个字段，包含 13 个属性，和一个房价的平均值。

下载数据集：

由于波士顿房价数据集是 Keras 中的内置数据集，可以 `keras.datasets` 模块访问数据集，使用 `load_data()` 方法加载数据集。

```
boston_housing = tf.keras.datasets.boston_housing
(train_x, train_y), (test_x, test_y) = boston_housing.load_data(test_split=0)
```

## 4 案例要求

观察波士顿房价数据集中各个属性对房价的影响，从中选择你认为对房价影响最明显的属性并说明理由；选择影响最明显的两个属性，训练二元线性回归模型，并测试其性能，以可视化的形式展现训练测试的过程。

(1) 选择属性；

(2) 根据(1)中的属性，编写代码，建立、训练并测试模型。并以可视化的形式展现训练和测试的过程及结果；

(3) 尝试调试学习率、迭代次数等超参数，使模型达到最优的性能，请记录超参数的调试过程及结果，并简要分析和总结；

(4) 尝试使用波士顿房价数据集中的其他属性，训练模型；

(5) 分析结果：比较分别使用单一属性和你自己选择的属性组合训练模型时，学习率、迭代次数等参数，以及在训练集和测试集上的均方差损失和模型训练时间等结果，以表格或其他合适的图表形式展示。通过以上结果，可以得到什么结论，或对你有什么启发。

## A3.3 波士顿房价预测：多元线性回归应用

本案例使用 TensorFlow 实现梯度下降法，求解多元线性回归，实现对波士顿房价预测。

### 1 学习目标

- ① 了解 Variable 可训练变量；
- ② 掌握使用 TensorFlow 自动求导机制实现梯度下降法；
- ③ 掌握梯度下降法求解多元线性回归的方法；
- ④ 掌握使用多维数组广播运算实现归一化方法；
- ⑤ 能够使用 TensorFlow 的可训练变量和自动求导机制实现梯度下降法，求解多元线性回归问题，实现对波士顿房价预测。

### 2 基础知识

#### 2.1 Variable 可训练变量

在学习 TensorFlow 的自动求导之前，先要知道什么是 Variable 对象。Variable 对象其实是对 tensor 对象的进一步封装，它在模型训练过程中自动记录梯度信息，并且由算法自动调整和优化它的取值。它是一个可以被训练的变量，在机器学习中作为模型参数，可以通过 assign()、assign\_add()、assign\_sub() 等函数对它进行赋值。

#### 2.2 TensorFlow 自动求导机制

TensorFlow 提供了一个专门用来求导的类 GradientTape，可以形象的理解为记录梯度数据的磁带。通过他，可以实现对变量的自动求导和监视。GradientTape 类实现了上下文管理器，它能够监视 with 语句块中所有的变量和计算过程，并把它们自动记录在梯度带中。以下是 GradientTape 类的构造函数。

```
With GradientTape() as tape:  
    函数表达式  
grad =tape.gradient(函数, 自变量)
```

#### 2.3 模型参数更新算法

使用梯度下降法求解多元线性回归，使用模型参数更新算法：

$$\frac{\partial Loss}{\partial W} = X^T(XW - Y)$$
$$W^{(k+1)} = W^{(k)} - \eta \frac{\partial Loss(W)}{\partial W}$$
$$W^{(k+1)} = W^{(k)} - \eta X^T(XW - Y)$$

其中，梯度信息使用 TensorFlow 自动求导机制对损失函数求导。

### 3 波士顿房价数据集

波士顿房价数据集取自卡内基梅隆大学维护的 StatLib 库，1978 年开始统计，涵盖了麻省波士顿的 506 个不同郊区的房屋数据。

一共含有 506 条数据，404 条训练数据集，102 条测试数据集。每条数据 14 个字段，包含 13 个属性，和一个房价的平均值。

下载数据集：

由于波士顿房价数据集是 Keras 中的内置数据集，可以 `keras.datasets` 模块访问数据集，使用 `load_data()`方法加载数据集。

```
boston_housing = tf.keras.datasets.boston_housing
(train_x, train_y), (test_x, test_y) = boston_housing.load_data(test_split=0)
```

#### 4 案例要求

使用波士顿房价数据集中的所有属性，训练多元线性回归模型，并测试其性能，以可视化的形式展现训练和测试的过程及结果。

(1) 编写代码，建立、训练并测试模型。并以可视化的形式展现训练和测试的过程及结果：

(2) 可视化输出预测房价与实际房价对比；

(3) 尝试调试学习率、迭代次数等超参数，使模型达到最优的性能，请记录超参数的调试过程及结果，并简要分析和总结；

(4) 分析结果：在 A3.1, A3.2 中，分别选择了单一属性和自选属性组合，比较分别使用单一属性、全部属性和你自己选择的属性组合训练模型时，学习率、迭代次数等参数，以及在训练集和测试集上的均方差损失和模型训练时间等结果，以表格或其他合适的图表形式展示。通过以上结果，可以得到什么结论，或对你有什么启发。

## A4.1 自建数据集商品房分类：一元逻辑回归应用

本案例使用一元逻辑回归模型实现对商品房分类。

### 1 学习目标

- ① 了解分类器的概念；
- ② 理解广义线性回归模型；
- ③ 掌握一元逻辑回归的基本原理；
- ④ 掌握 Sigmoid 函数、交叉熵损失函数的实现方法；
- ⑤ 能够使用一元逻辑回归实现对商品房分类。

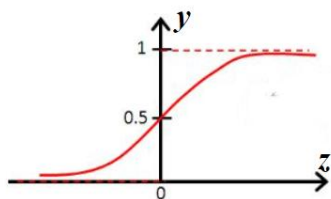
### 2 基础知识

#### 2.1 逻辑回归

logistic 回归是一种广义线性回归（generalized linear model），是分类模型，常用于二分类。也是线性模型的一种，由线性回归 + Sigmoid 函数组成，使用 Sigmoid 函数作为广义线性回归中的联系函数。逻辑回归使用线性回归的结果，作为对数几率函数的自变量，实现一个分类器，不仅可以预测出类别，还可以预测出输入样本属于某个类别的概率。

#### 2.2 Sigmoid 函数

Sigmoid 函数是一个 S 型函数，也称为 S 型生长曲线。在深度学习中，Sigmoid 函数常被用作神经网络的激活函数。Sigmoid 能够将变量从负无穷大到正无穷大的输入转化为[0,1]之间的值，表示概率。公式如下：



$$y = g^{-1}(z) = \sigma(z) = \sigma(wx + b)$$
$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(wx+b)}}$$

#### 2.3 交叉熵损失函数

在逻辑回归中，使用交叉熵函数计算预测值与标签值之间的误差，交叉熵损失函数可以解决平方损失函数应用于逻辑回归时，参数更新过慢或陷入局部极小值的问题。

公式如下，其中 $\hat{y}_i$ 是预测概率：

$$Loss = -\sum_{i=1}^n [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)]$$

$y_i$	第 <i>i</i> 个样本的标记
$\hat{y}_i$	$\hat{y}_i = \sigma(wx_i + b)$

### 3 数据集

使用下表中的“商品房记录表”作为样本数据，使用房屋面积和商品房类型，训练一元线性回归模型，实现一个商品房分类。（0 代表普通住宅，1 代表高档住宅）



表 A4.1.1 商品房销售记录表

序号	面积 (平方米)	类型	序号	面积 (平方米)	类型
1	137.97	1	9	106.69	0
2	104.50	1	10	140.05	1
3	100.00	0	11	53.75	0
4	126.32	1	12	46.91	0
5	79.20	0	13	68.00	0
6	99.00	1	14	63.02	0
7	124.00	1	15	81.26	0
8	114.00	0	16	86.21	0

#### 4 案例要求

将“商品房记录表”作为样本数据，使用房屋面积和类型，训练多元线性回归模型，实现商品房类型分类。

要求：

- (1) 编写代码，建立、训练并测试模型，并以可视化的形式展现训练和测试的过程及结果；
- (2) 根据每轮训练的权值绘制 Sigmoid 函数，更好体现模型训练过程；
- (3) 尝试调试超参数，使模型达到最优的性能，记录实验过程和结果；
- (4) 分析结果：通过以上结果，可以得到什么结论，或对你有什么启发。

## A4.2 鸢尾花二分类：多元逻辑回归应用

本案例使用多元逻辑回归模型实现对鸢尾花数据集的二分类。

### 1 学习目标

- ① 理解广义线性回归模型；
- ② 理解二分类任务；
- ③ 掌握多元逻辑回归的基本原理；
- ④ 掌握 Sigmoid 函数、交叉熵损失函数的实现方法；
- ⑤ 掌握绘制分类图的方法；
- ⑥ 能够使用逻辑回归实现对鸢尾花的二分类，并根据分类模型，绘制分类图。

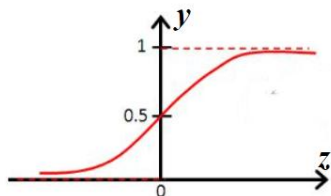
### 2 基础知识

#### 2.1 逻辑回归

logistic 回归是一种广义线性回归（generalized linear model），是分类模型，常用于二分类。也是线性模型的一种，由线性回归 + Sigmoid 函数组成，使用 Sigmoid 函数作为广义线性回归中的联系函数。逻辑回归使用线性回归的结果，作为对数几率函数的自变量，实现一个分类器，不仅可以预测出类别，还可以预测出输入样本属于某个类别的概率。

#### 2.2 Sigmoid 函数

Sigmoid 函数是一个 S 型函数，也称为 S 型生长曲线。在深度学习中，Sigmoid 函数常被用作神经网络的激活函数。Sigmoid 能够将变量从负无穷大到正无穷大的输入转化为[0,1]之间的值，表示概率。公式如下：



$$y = g^{-1}(z) = \sigma(z) = \sigma(wx + b)$$
$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(wx+b)}}$$

在多元模型中，公式如下：

$$y = \frac{1}{1 + e^{-(w^T X)}}$$

其中,  $W = (w_0, w_1, \dots, w_m)^T$ ,  $X = (x^0, x^1, \dots, x^m)^T$ ,  $x^0=1$ 。

#### 2.3 交叉熵损失函数

在逻辑回归中，使用交叉熵函数计算预测值与标签值之间的误差，交叉熵损失函数可以解决平方损失函数应用于逻辑回归时，参数更新过慢或陷入局部极小值的问题。

公式如下，其中  $\hat{y}_i$  是预测概率：

$$Loss = -\sum_{i=1}^n [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)]$$

$y_i$	第 <i>i</i> 个样本的标记
$\hat{y}_i$	$\hat{y}_i = \sigma(wx_i + b)$

### 3 Iris 数据集

Iris 数据集是常用的分类实验数据集，由 Fisher, 1936 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（Setosa, Versicolour, Virginica）三个种类中的哪一类。

下载数据集：

通过 keras.utils 模块，使用 get\_file() 函数进行下载，read\_csv() 读取数据。

```
TRAIN_URL = "http://download.tensorflow.org/data/iris_training.csv"
train_path=tf.keras.utils.get_file(TRAIN_URL.split('/')[1],TRAIN_URL)
TEST_URL = "http://download.tensorflow.org/data/iris_test.csv"
test_path=tf.keras.utils.get_file(TEST_URL.split('/')[1],TEST_URL)
```

### 4 案例要求

观察案例 A1.2 鸢尾花数据集可视化结果(如图 A4.2.1 所示)，选择恰当的属性或属性组合，训练逻辑回归模型，区分山鸢尾和变色鸢尾。记录和分析实验结果，并给出总结。

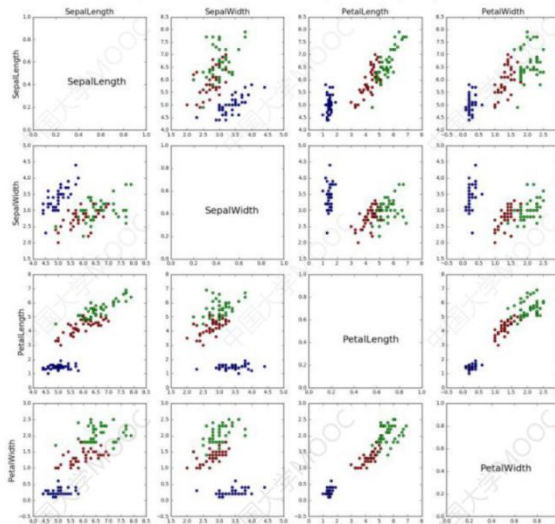


图 A4.2.1 鸢尾花数据集可视化

- (1) 确定属性选择方案；
- (2) 编写代码，建立、训练并测试模型。并以可视化的形式展现训练和测试的过程及结果；
- (3) 选择不同属性或属性组合对模型进行训练，比较选择不同属性组合时的实验结果，确定区分两种类别的最佳属性组合；
- (4) 根据确定的属性组合，尝试调试超参数，综合考虑准确率，交叉熵损失和训练时间，使模型在测试集达到最优的性能；
- (5) 尝试修改模型，区分山鸢尾和维吉尼亚鸢尾；
- (6) 分析结果：比较选择不同属性组合以及超参数时的实验结果，确定区分山鸢尾和变色鸢尾，至少需要几种属性？说明选择某种属性或属性组合的依据；通过以上结果，可以得到什么结论，或对你有什么启发。

## A4.3 鸢尾花三分类：Softmax 回归应用

本案例使用 Softmax 回归模型实现对鸢尾花数据集的分类。

### 1 学习目标

- ① 理解多分类任务；
- ② 理解 Softmax 回归的基本原理；
- ③ 掌握独热编码、Softmax 函数以及交叉熵损失函数的实现方法；
- ④ 掌握多分类任务中准确率的计算方法；
- ⑤ 掌握绘制分类图的方法；
- ⑥ 能够使用 Softmax 回归实现对鸢尾花的分类。

### 2 基础知识

#### 2.1 softmax 回归

Softmax 回归用于解决多分类任务，当输入数据可分为多个类别时，Softmax 回归估算输入数据归属于每一类的概率。

#### 2.2 独热编码

One-Hot 编码，又称为一位有效编码，主要是采用 N 位状态寄存器来对 N 个状态进行编码，每个状态都由他独立的寄存器位，并且在任意时候只有一位有效。也就是说该离散型类别特征有 N 个不同的取值，就有 N 个状态，就用 N 位寄存器（N 位二进制）来表示 N 个不同的值。每 N 个比特位表示的值只有一个是 1，其余为 0，其实就是以二进制的形式表示，主要用于特征内部取值是无序的。

直观来说就是有多少个状态就有多少比特，而且只有一个比特为 1，其他全为 0 的一种码制，能够更加合理的表示数据之间的关系，有效避免学习过程中的偏差。将多分类问题中的类别标签，采用独热编码方式来表示，使非偏序关系的数据，取值不具有偏序性。

#### 2.3 Softmax()函数

Softmax 的含义就在于不再唯一的确定某一个最大值，而是为每个输出分类的结果都赋予一个概率值，表示属于每个类别的可能性，完成分类任务。

下面给出 Softmax 函数的定义（以第 k 个节点输出为例）：

$$y_k = \frac{e^{z_k}}{\sum_{p=1}^c e^{z_p}}$$

其中  $Z_k$  为第 k 个节点的输出值，C 为输出节点的个数，即分类的类别个数。通过 Softmax 函数就可以将多分类的输出值转换为范围在 [0, 1] 和为 1 的概率分布。

#### 2.4 多分类交叉熵损失函数

在多分类问题中，使用多分类交叉熵函数计算预测值与标签值之间的误差，公式如下：

$$Loss = -\sum_{i=1}^n \sum_{p=1}^c y_{i,p} \ln(\hat{y}_{i,p})$$

### 3 Iris 数据集

Iris 数据集是常用的分类实验数据集，由 Fisher, 1936 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（Setosa, Versicolour, Virginica）三个种类中的哪一类。

下载数据集：

通过 `keras.utils` 模块，使用 `get_file()` 函数进行下载，`read_csv()` 读取数据。

```
TRAIN_URL = "http://download.tensorflow.org/data/iris_training.csv"
train_path=tf.keras.utils.get_file(TRAIN_URL.split('/')[1],TRAIN_URL)
TEST_URL = "http://download.tensorflow.org/data/iris_test.csv"
test_path=tf.keras.utils.get_file(TEST_URL.split('/')[1],TEST_URL)
```

### 4 案例要求

参考案例“A1.2 鸢尾花数据集可视化”结果，分别选择 2 种、3 种和 4 种属性，编写程序，区分三种鸢尾花。记录和分析实验结果，并给出总结。

- (1) 确定属性选择方案；
- (2) 编写代码建立、训练并测试模型；
- (3) 绘制分区图，可视化输出分类结果；
- (4) 尝试调试超参数，综合考虑准确率、交叉熵损失、和训练时间等，使模型在测试集达到最优的性能；
- (5) 分析结果： 比较选择不同属性组合以及超参数时的实验结果，设计表格或合适的图表展示；分析实验结果，可以得到什么结论，或者对你有什么启发。

## A4.4 鸢尾花三分类：单层神经网络应用

本案例使用单层前馈型神经网络模型实现对鸢尾花数据集的分类。

### 1 学习目标

- ① 了解感知机训练法则；
- ② 理解 Delta 法则的基本原理；
- ③ 掌握单层神经网络模型结构的设计方法；
- ④ 掌握输入数据、模型参数与输出数据之间的运算关系；
- ⑤ 能够使用单层前馈性神经网络实现对鸢尾花的分类。

### 2 基础知识

#### 2.1 感知机训练法则

为了得到可接受的权向量，感知器法则是从随机的权值开始，然后反复地对每一个训练样例应用这个感知器，在当前感知器误分类样例时修改感知器的权值。重复这个过程，直到感知器正确分类所有的训练样例。修改权值的法则如下：

$$w_i^{(k+1)} = w_i^{(k)} + \Delta w_i$$
$$\Delta w_i = \eta (y - \hat{y}) x_i$$

这里， $w_i$ 为每次迭代时  $x_i$  对应的权值， $\Delta w_i$ 为每次  $w_i$  的修正值， $y$  是当前训练样本的目标输出， $\hat{y}$ 是感知器的输出， $\eta$ 是学习速率。训练样例线性可分时，在经过有限次地使用感知器训练法则后，上面的训练过程会收敛到一个正确分类所有训练样例的权向量；如果数据不是线性可分的，那么不能保证训练过程收敛。

#### 2.2 Delta 法则

当训练样例线性可分时,感知器法则可以成功地找到一个权向量,但当样例是非线性可分的数据集时,感知机训练法则无法收敛。为了克服感知及训练法则,对非线性数据集无法收敛的问题,提出 Delta 法则。Delta 关键思想是使用梯度下降法,找到能够最佳拟合训练样本集的权向量

#### 2.3 单层前馈型神经网络

前馈网络指每一层神经元接收前一层神经元的输出,并输出到下一层神经元。可以看作一个函数,通过简单非线性函数的多次复合,实现输入空间到输出空间的复杂映射。

单层前馈神经网络是最简单的一种人工神经网络,其只包含一个输出层,输出层上节点的值(输出值)通过输入值乘以权重值直接得到。

### 3 Iris 数据集

Iris 数据集是常用的分类实验数据集,由 Fisher, 1936 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 (Setosa, Versicolour, Virginica) 三个种类中的哪一类。

下载数据集:

通过 `keras.utils` 模块,使用 `get_file()`函数进行下载, `read_csv()`读取数据。

```
TRAIN_URL = "http://download.tensorflow.org/data/iris_training.csv"
train_path=tf.keras.utils.get_file(TRAIN_URL.split('/')[-1],TRAIN_URL)
TEST_URL = "http://download.tensorflow.org/data/iris_test.csv"
test_path=tf.keras.utils.get_file(TEST_URL.split('/')[-1],TEST_URL)
```

#### 4 案例要求

使用单层神经网络，实现对 Iris 数据集中的三种鸢尾花分类，并测试模型性能，以恰当的形式展现训练过程和结果。

- (1) 设计神经网络，确定神经网络的结构、激活函数和损失函数；
- (2) 编写代码，建立、训练并测试模型，输出训练时间、测试时间、损失与准确率曲线等；
- (3) 尝试改变超参数，综合考虑准确率、交叉熵损失、和训练时间等，使模型在测试集达到最优的性能，并以恰当的方式记录和展示实验结果；
- (4) 分析结果：简要说明你寻找最佳超参数的过程，请分析它们对结果准确性和训练时间的影响，以表格或其他合适的图表形式展示。通过以上结果，可以得到什么结论，或对你有什么启发。

## A4.5 鸢尾花三分类：多层神经网络应用

本案例使用多层神经网络模型实现对鸢尾花数据集的分类。

### 1 学习目标

- ① 了解常用的激活函数；
- ② 理解并掌握误差反向传播算法；
- ③ 掌握多层神经网络结构的设计方法；
- ④ 掌握 ReLU 函数的基本原理；
- ⑤ 能够使用多层神经网络实现对鸢尾花的分类。

### 2 基础知识

#### 2.1 误差反向传播算法

反向传播算法（back propagation，简称 BP 模型）是 1986 年由 Rumelhart 和 McClelland 为首的科学家提出的概念，是一种按照误差逆向传播算法训练的多层前馈神经网络，是目前应用最广泛的神经网络。

利用链式法则，反向传播损失函数的梯度信息，计算出损失函数对网络中所有模型的梯度。其基本思想是，学习过程由正向传播与误差的反向传播两个过程组成：

(1) 正向传播：输入样本→输入层→各隐含层（处理）→输出层

注 1：若输出层实际输出与期望输出不符，则转入 2）（误差反向传播过程）。

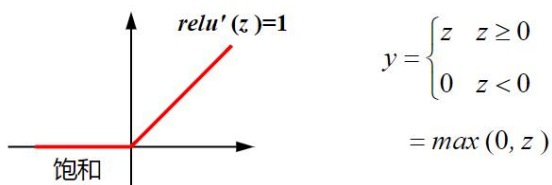
(2) 误差反向传播：输出误差（某种形式）→隐含层（逐层）→输入层，其主要目的是通过将输出误差反传，将误差分摊给各层所有单元，从而获得各层单元的误差信号，进而修正各单元的权值。

#### 2.2 ReLU 函数

深度学习中最大的问题是梯度消失问题，使用 Tanh、Sigmoid 等激活函数情况下特别严重（神经网络在进行方向误差传播时，各个层都要乘以激活函数的一阶导数，梯度每传递一层就会衰减一层，网络层数较多时，梯度 G 就会不停衰减直到消失），使得训练网络收敛越来越慢，而 ReLU 函数凭借其线性、非饱和的形式，训练速度则快很多。

修正线性单元（Rectified Linear Unit, ReLU），又称线性整流函数，是一个分段函数：

当  $z$  大于 0，是线性函数，输出等于  $z$  本身， $z$  小于 0，输出均为 0。相对于 Sigmoid 的两



端饱和，ReLU 函数只有在零点的左边饱和，而在零点的右边，导数恒等于 1，缓解了梯度消失问题。

$z$  小于 0 时，梯度为 0，导致参数无法更新，神经元死亡，为解决 ReLU 函数的 ReLU 神经元死亡问题，还可以采用 Leaky-ReLU 函数、PReLU 函数（参数化修正线性单元）和 RReLU 函数（随机修正线性单元）。



## 2.3 多层神经网络的训练过程

在多层神经网络中，使用梯度下降法训练模型参数，通过误差反向传播算法计算梯度。多层神经网络的训练过程可以概括为，正向传递信号，反向传递误差。

首先，在输入层输入样本特征  $x$ ，在神经网络中逐层传递，直到在输出层得到预测值；然后，将神经网络得到的预测值与标签值比较，计算损失；若损失比较大，就使用梯度下降法调整最后一层神经元的参数，反向传播梯度信息，逐层后退，更新模型参数，完成一轮训练。

参数调整后的网络，再次根据样本特征，正向计算预测值，反向传播误差，调整模型参数，不断训练，直到标签值与网络输出一致。

## 3 Iris 数据集

Iris 数据集是常用的分类实验数据集，由 Fisher, 1936 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（Setosa, Versicolour, Virginica）三个种类中的哪一类。

下载数据集：

通过 `keras.utils` 模块，使用 `get_file()` 函数进行下载，`read_csv()` 读取数据。

```
TRAIN_URL = "http://download.tensorflow.org/data/iris_training.csv"
train_path=tf.keras.utils.get_file(TRAIN_URL.split('/')[1],TRAIN_URL)
TEST_URL = "http://download.tensorflow.org/data/iris_test.csv"
test_path=tf.keras.utils.get_file(TEST_URL.split('/')[1],TEST_URL)
```

## 4 案例要求

使用多层神经网络，实现对 Iris 数据集中的三种鸢尾花分类，并测试模型性能，以恰当的形式展现训练过程和结果。

(1) 设计神经网络，确定神经网络的隐含层层数和各层的节点数，各层中使用的激活函数，以及损失函数；

(2) 编写代码，建立、训练并测试模型，输出训练时间、测试时间、损失与准确率曲线等；

(3) 改变隐含层层数、隐含层中节点数等超参数，综合考虑准确率、交叉熵损失、和训练时间等，使模型在测试集达到最优的性能，并以恰当的方式记录和展示实验结果；

(4) 分析结果：简要说明你寻找最佳超参数的过程，请分析它们对结果准确性和训练时间的影响，以表格或其他合适的图表形式展示。通过以上结果，可以得到什么结论，或对你有什么启发。

## A5.1 手写数字识别：Keras.Sequential 模型应用

本案例使用 Keras.Sequential 模型实现手写数字识别。

### 1 学习目标

- ① 了解批量梯度下降、随机梯度下降和小批量梯度下降算法；
- ② 了解 Keras 和 tf.Keras；
- ③ 理解小批量梯度下降法的基本原理以及小批量梯度下降法的优化算法；
- ④ 掌握构建 Sequential 模型的方法；
- ⑤ 掌握模型训练的保存与加载方法；
- ⑥ 能够使用 Sequential 模型实现手写数字识别。

### 2 基础知识

#### 2.1 小批量梯度下降算法

批量梯度算法，每次迭代都使用所有样本来计算偏导数，由所有样本确定梯度方向，每一步都是准确地向着极值点趋近，迭代次数少，收敛于全局极小值或局部极小值点，可以利用向量运算进行并行计算，提升运算速度。计算量大，训练时间长，不适合大规模数据集。

随机批量算法，每次迭代只选择一个样本训练模型，使网络的输出尽可能逼近这个样本的标签值；使用所有样本训练一遍称为一轮，反复训练多轮，直到网络对所有样本的误差足够小。随机批量算法参数更新非常频繁，训练轮数非常多，无法快速收敛，不易于实现并行计算。

小批量梯度下降法是批量梯度算法和随机批量算法的折中方案，把数据分为多个小批量，每次迭代使用一个小批量来训练模型。每个小批量中的所有样本共同决定了本次迭代中梯度的方向，使用所有小批量训练一遍称为一轮，需要训练多轮，使网络对所有样本的误差足够小，每次迭代的训练样本数固定，与整个训练集的样本数量无关。可以实现并行计算，训练大规模数据集。

#### 2.2 梯度下降法的优化算法

对于多层神经网络，使用梯度下降法，从理论上无法保证一定可以达到最小值点，只能尽量改进训练方法，调整参数，优化算法，使它尽可能收敛于全局最小值。

影响小批量下降算法的主要因素有：小批量样本的选择、批量大小、学习率、梯度。可以从这四个方面优化算法。

小批量样本：在每轮训练前，打乱样本顺序。

批量大小：尽可能选择合适的批量大小，使得模型效率更高，更稳定，并且具有比较好的泛化能力。

学习率：(1) 凸函数：使用学习率衰减算法

(2) 非凸函数：自适应学习率调节算法，如 AdaGrad、RMSprop、AdaDelta。

梯度：采用动量梯度下降法（Momentum）、牛顿加速梯度算法（NAG）。

#### 2.3 tf.Keras

tf.Keras 是 TensorFlow 的高阶 API，通过 tf.Keras 可以快速搭建和训练神经网络模型，Keras 中的主要数据结构是模型。

## 2.4 Sequential 模型

Sequential 是 Keras 中的一种神经网络框架，可以被认为是一个容器，其中封装了神经网络的结构。Sequential 模型只有一组输入和一组输出。各层之间按照先后顺序进行堆叠。前面一层的输出就是后面一次的输入。通过不同层的堆叠，构建出神经网络。

## 3 MNIST 数据集

MNIST 数据集由 Yann LeCun 搜集，是一个大型的手写体数字数据库，通常用于训练各种图像处理系统，也被广泛用于机器学习领域的训练和测试。MNIST 数字识别数据集数据量不会太多，而且是单色的图像，较简单，适合深度学习初学者练习建立模型、训练、预测。

MNIST 数据集共有训练数据 60000 项、测试数据 10000 项。数据集是有 250 人手写数字组成，一半是高中生，一半是美国人口普查局。每张图像的大小为 28\*28（像素），每张图像都为灰度图像，位深度为 8（灰度图像是 0-255）。但并不是有以图片形式存储的，而是存储在一个 28\*28 的二维数组中，数组中的每个元素对应于图片中的一个像素。MNIST 数据集已经被集成在 Keras 中了，可以直接使用 Keras 中的 datasets 模块访问。

下载数据集：

通过 keras 模块中的 datasets 模块访问：

```
mnist = tf.keras.datasets.mnist
(train_x, train_y), (test_x, test_y) = mnist.load_data()
```

## 4 案例要求

1、下载 MNIST 数据集，使用神经网络模型，实现对 MNIST 手写数字数据集的识别，并测试模型性能，记录和分析结果。

(1) 设计神经网络，确定神经网络的结构、激活函数和损失函数；

(2) 编写代码，建立、训练并测试模型；

(3) 尝试改变超参数，调整超参数和训练参数，使模型在测试集达到最优的性能，并以恰当的方式记录和展示实验过程和结果；

(4) 保存上述训练好的模型；

(5) 分析结果：对实验结果进行分析，说明神经网络的宽度和深度对模型性能的影响，以及优化器、损失函数等对模型性能的影响。通过以上结果，可以得到什么结论，或对你有什么启发。

2、编写程序，使用训练好的模型，实现对自制手写数字数据集的识别。

要求：

(1) 请下载附件 1 中的“自制手写数字-a 组.zip”，加载 1 中训练好的模型，对“自制手写数字-a 组”中的手写数字进行识别，记录并分析结果。

(2) 请下载附件 1 中的“自制手写数字-b 组.zip”，加载 1 中训练好的模型，对“自制手写数字-b 组”文件夹中的手写数字进行识别，记录并分析结果。

(3) 请下载附件 1 中的“自制手写数字-c 组.zip”，将其中的手写数字转换为 28×28 的数组形式。加载 1 中训练好的模型，对其进行识别，记录并分析结果。

(4) 对比 3 组图像的识别准确率，对于识别率偏低的测试组，分析原因，并思考应如何改进。

提示:

①缩放图像: `img.resize()`

②二值化图像: `img.convert('1', dither=Image.NONE)`, 此处的 `dither` 参数是防止像素点抖动

③反色处理: `img_reverse=1.0 - asarray(img).astype(np.int) / 1.0001`

④裁剪图像: `img.crop()`

附件 1 手写数字图像:

链接:<https://pan.baidu.com/s/1bagitzF-MjIp1CN3DImHVg?pwd=1234>

提取码:1234

## A5.2 手写数字识别：卷积神经网络应用

本案例使用卷积神经网络模型实现手写数字识别。

### 1 学习目标

- ① 理解深度学习的基本原理；
- ② 理解图像识别的基本原理；
- ③ 掌握图像卷积运算的方法；
- ④ 掌握卷积神经网络的基本思想；
- ⑤ 掌握卷积神经网络的构建方法；
- ⑥ 能够使用 TensorFlow 实现卷积神经网络，完成手写数字识别。

### 2 基础知识

#### 2.1 深度学习

深度学习是机器学习的一种，而机器学习是实现人工智能的必经路径。深度学习的概念源于人工神经网络的研究，含多个隐藏层的多层感知器就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。研究深度学习的动机在于建立模拟人脑进行分析学习的神经网络，它模仿人脑的机制来解释数据，例如图像，声音和文本等。

#### 2.2 图像识别

图像识别是指利用计算机对图像进行处理和分析，使机器能够理解图像中的内容。图像识别算法是计算机视觉中非常重要且基础的分支，类似于人类对图像内容的识别其主要任务是通过图像中像素分布及颜色、纹理等特征的统计，将图像内容所属类别进行正确的分类。在深度学习中，图像识别模型在完成本职任务的同时还充当计算机视觉其他任务的特征提取网络。

#### 2.3 卷积运算

卷积运算就是对于图像  $f(x,y)$  中的每个像素，将其作为中心像素，计算它及其邻域像素和 kernel 核  $w(a,b)$  对应位置元素的乘积，然后把结果相加到一起，得到的值就作为该中心像素的新值，这样就完成了一次卷积运算。然后将 kernel 向下或向左平移一位继续计算每个像素的新值，直到遍历完整个图像。实现模糊 (blurring)，锐化 (sharpening)，边缘检测 (edge detection) 等功能。

#### 2.4 卷积神经网络

卷积神经网络 (Convolutional Neural Networks, CNN) 是一类包含卷积计算且具有深度结构的前馈神经网络 (Feedforward Neural Networks)，是深度学习 (deep learning) 的代表算法之一。它使用卷积核 (Convolutional kernel) 从数据中提取重要信息，使用图像填充的策略尽量保留原始数据的信息，使用池化等策略来避免过拟合。

卷积神经网络主要包含三层，卷积层 (Convolutional Layer) 主要作用是提取特征。池化层 (Max Pooling Layer) 主要作用是下采样 (downsampling)，在减少数据处理量的同时，保留有用信息。全连接层 (Fully Connected Layer) 主要作用是分类。

卷积神经网络采用局部连接与权值共享机制，使得网络的结构更接近于实际的生物神经网络，降低了网络的复杂性，模型参数的数量远小于全连接网络，而且由于同一层中的神经

元权值相同，网络可以并行学习；同时卷积核中的权值是从数据中心学习得到的，因此更加灵活。

### 3 MNIST 数据集

MNIST 数据集由 Yann LeCun 搜集，是一个大型的手写体数字数据库，通常用于训练各种图像处理系统，也被广泛用于机器学习领域的训练和测试。MNIST 数字识别数据集数据量不会太多，而且是单色的图像，较简单，适合深度学习初学者练习建立模型、训练、预测。

MNIST 数据集共有训练数据 60000 项、测试数据 10000 项。数据集是有 250 人手写数字组成，一半是高中生，一半是美国人口普查局。每张图像的大小为 28\*28（像素），每张图像都为灰度图像，位深度为 8（灰度图像是 0-255）。但并不是有以图片形式存储的，而是存储在一个 28\*28 的二维数组中，数组中的每个元素对应于图片中的一个像素。MNIST 数据集已经被集成在 Keras 中了，可以直接使用 Keras 中的 datasets 模块访问。

下载数据集：

通过 keras 模块中的 datasets 模块访问：

```
mnist = tf.keras.datasets.mnist
(train_x, train_y), (test_x, test_y) = mnist.load_data()
```

### 4 案例要求

使用 Keras 构建和训练卷积神经网络，实现对 Mnist 手写数字数据集的识别，并测试模型性能，以恰当的形式展现训练过程和结果。

要求：

- (1) 设计卷积神经网络，确定卷积神经网络的结构、激活函数和损失函数；
- (2) 编写代码，构建卷积神经网络，实现上述功能；
- (3) 调整超参数，记录实验过程和结果。调整卷积神经网络的结构和训练参数，找出最佳的结构和超参数，记录和分析实验结果；
- (4) 保存最佳模型，计算各层参数个数和模型总参数；
- (5) 使用模型对手写数字进行预测，并可视化输出结果；
- (6) 分析结果：对实验结果进行分析，说明卷积神经网络的超参数有哪些，以及它们对模型性能的影响。通过以上结果，可以得到什么结论，或对你有什么启发。

## A6.1 Cifar10 图像识别：卷积神经网络的优化应用

本案例使用优化的卷积神经网络模型实现 cifar10 数据集图像识别。

### 1 学习目标

- ① 了解 Cifar10 数据集；
- ② 理解图像识别的基本原理；
- ③ 掌握图像卷积运算的方法；
- ④ 掌握卷积神经网络的优化方法；
- ⑤ 掌握卷积神经网络的构建方法；
- ⑥ 能够使用 TensorFlow 实现卷积神经网络，完成 cifar10 图像识别。

### 2 基础知识

#### 2.1 深度学习

深度学习是机器学习的一种，而机器学习是实现人工智能的必经路径。深度学习的概念源于人工神经网络的研究，含多个隐藏层的多层感知器就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。研究深度学习的动机在于建立模拟人脑进行分析学习的神经网络，它模仿人脑的机制来解释数据，例如图像，声音和文本等。

#### 2.2 图像识别

图像识别是指利用计算机对图像进行处理和分析，使机器能够理解图像中的内容。图像识别算法是计算机视觉中非常重要且基础的分支，类似于人类对图像内容的识别其主要任务是通过图像中像素分布及颜色、纹理等特征的统计，将图像内容所属类别进行正确的分类。在深度学习中，图像识别模型在完成本职任务的同时还充当计算机视觉其他任务的特征提取网络。

#### 2.3 卷积运算

卷积运算就是对于图像  $f(x,y)$  中的每个像素，将其作为中心像素，计算它及其邻域像素和 kernel 核  $w(a,b)$  对应位置元素的乘积，然后把结果相加到一起，得到的值就作为该中心像素的新值，这样就完成了一次卷积运算。然后将 kernel 向下或向左平移一位继续计算每个像素的新值，直到遍历完整个图像。实现模糊 (blurring)，锐化 (sharpening)，边缘检测 (edge detection) 等功能。

#### 2.4 卷积神经网络

卷积神经网络 (Convolutional Neural Networks, CNN) 是一类包含卷积计算且具有深度结构的前馈神经网络 (Feedforward Neural Networks)，是深度学习 (deep learning) 的代表算法之一。它使用卷积核 (Convolutional kernel) 从数据中提取重要信息，使用图像填充的策略尽量保留原始数据的信息，使用池化等策略来避免过拟合。

卷积神经网络主要包含三层，卷积层 (Convolutional Layer) 主要作用是提取特征。池化层 (Max Pooling Layer) 主要作用是下采样 (downsampling)，在减少数据处理量的同时，保留有用信息。全连接层 (Fully Connected Layer) 主要作用是分类。

卷积神经网络采用局部连接与权值共享机制，使得网络的结构更接近于实际的生物神经网络，降低了网络的复杂性，模型参数的数量远小于全连接网络，而且由于同一层中的神经

元权值相同，网络可以并行学习；同时卷积核中的权值是从数据中心学习得到的，因此更加灵活。

### 2.5 卷积神经网络的优化

Dropout()可以比较有效的缓解过拟合的发生，在一定程度上达到正则化的效果。在神经网络的训练过程中，对于一次迭代中的某一层神经网络，先随机选择中的一些神经元并将其临时隐藏(丢弃)，然后再进行本次训练和优化。在下次迭代中，继续随机隐藏一些神经元，如此直至训练结束。由于是随机丢弃，故而每一个 mini-batch 都在训练不同的网络。使用 Dropout 使模型的泛化性能更强，节省了模型训练时间。

级联卷积核，通过堆叠多个小卷积核代替一个大的卷积核，使用级联卷积核不仅所需要的参数变少，加入的非线性变换也越多，使模型的非线性表达能力更好。

### 3 Cifar-10 数据集

Cifar-10 是由 Hinton 的学生 Alex Krizhevsky、Ilya Sutskever 收集的一个用于普适物体识别的计算机视觉数据集。由 60000 张 32x32 彩色图片组成，这个数据集涵盖了 10 个类别（飞机，汽车，鸟，猫，鹿，狗，青蛙，马，船以及卡车）。每种类型有 6000 张图片，训练集有 50000 张图片，测试集有 10000 张图片。

Cifar-10 是 Keras 中集成的数据集，可以直接使用 Keras 中的 datasets 模块访问数据集。

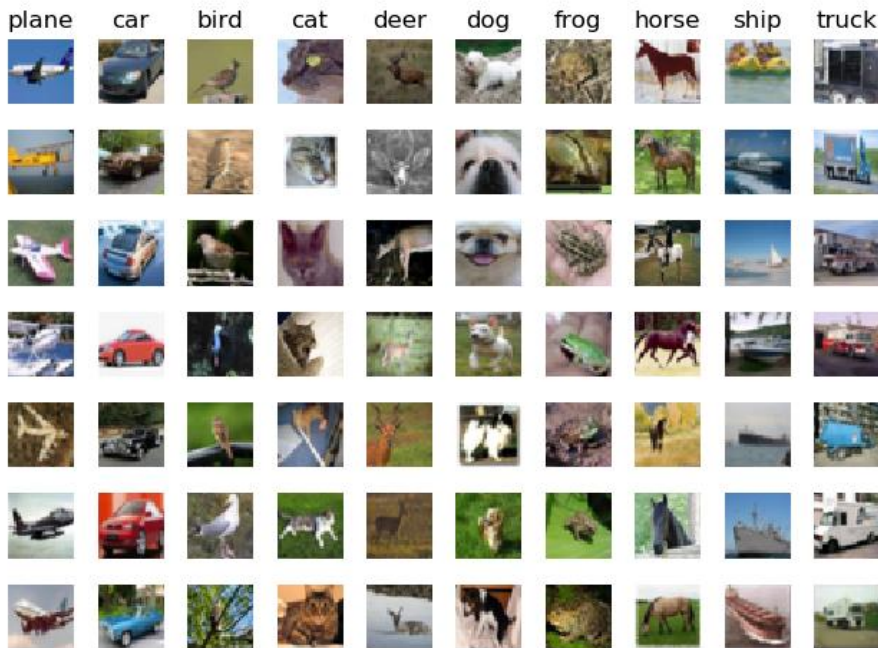


图 A6.1.1 Cifar-10 部分数据集

下载数据集：

通过 keras 模块中的 datasets 模块访问：

```
cifar10=tf.keras.datasets.cifar10
(x_train,y_train),(x_test,y_test)=cifar10.load_data()
```



## 4 案例要求

使用 Keras 构建和训练卷积神经网络，实现对 Cifar10 数据集的识别，并测试模型性能，以恰当的形式展现训练过程和结果。

要求：

- (1) 设计卷积神经网络，确定卷积神经网络的结构、激活函数和损失函数；
- (2) 编写代码，构建卷积神经网络，实现上述功能；
- (3) 调整超参数，记录实验过程和结果。调整卷积神经网络的结构和训练参数，找出最佳的结构和超参数，记录和分析实验结果；
- (4) 保存最佳模型，计算各层参数个数和模型总参数；
- (5) 使用模型对 Cifar10 图像进行预测，并可视化输出结果；
- (6) 分析结果：对实验结果进行分析，说明卷积神经网络的超参数有哪些，以及它们对模型性能的影响。通过以上结果，可以得到什么结论，或对你有什么启发。