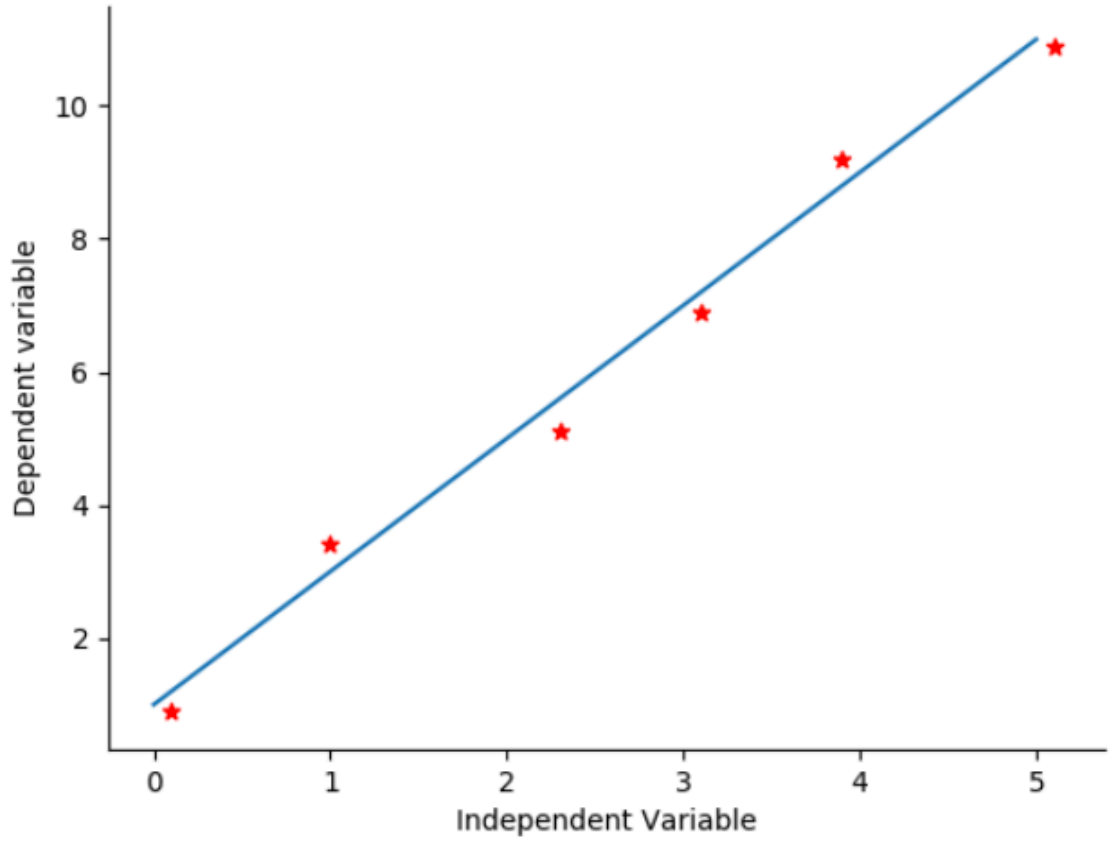
# 1.线性回归分析（ Linear Regression Analysis）

**线性回归分析（Regression Analysis）**：其数据集是给定一个函数和他的一些坐标点，然后通过回归分析的算法，来估计原函数的模型，求得最符合这些数据集的函数解析式。然后我们就可以用来预估未知数据，输入一个自变量便会根据这个模型解析式输出因变量，这些自变量就是特征向量，因变量即为标签，而且标签的值是建立在连续范围内的。 通俗来讲就是我们在做数学题的时候，解未知数的方法。假如给定自变量和函数，通过函数处理自变量，然后获得函数的解。而回归分析便是相当于给定自变量和函数的解，然后去求函数。如下图所示，我们已经知道红色点坐标，然后回归得到直线，回归分析属于**监督学习**。




上图只是简单的一元线性分析，回归后我们可以得到如$f(x) = a * x + b$的函数表达式，但更多情况下我们是求解多元线性回归问题，那应该如何解决呢。

# 2.模型表达

建立数学模型之前，我们先定义如下变量。

- $x_i$表示输入数据（Feature）
- $y_i$表示输出数据（Target）
- $(x_i, y_i)$表示一组训练数据（Training example）
- m表示训练数据的个数
- n表示特征数量

监督学习目标便是根据给定的训练数据，可以得到函数方法，使得假设函数$h$(hypothesis)满足$h(x) -> y$。针对线性回归而言，函数$h(x)$表达式为

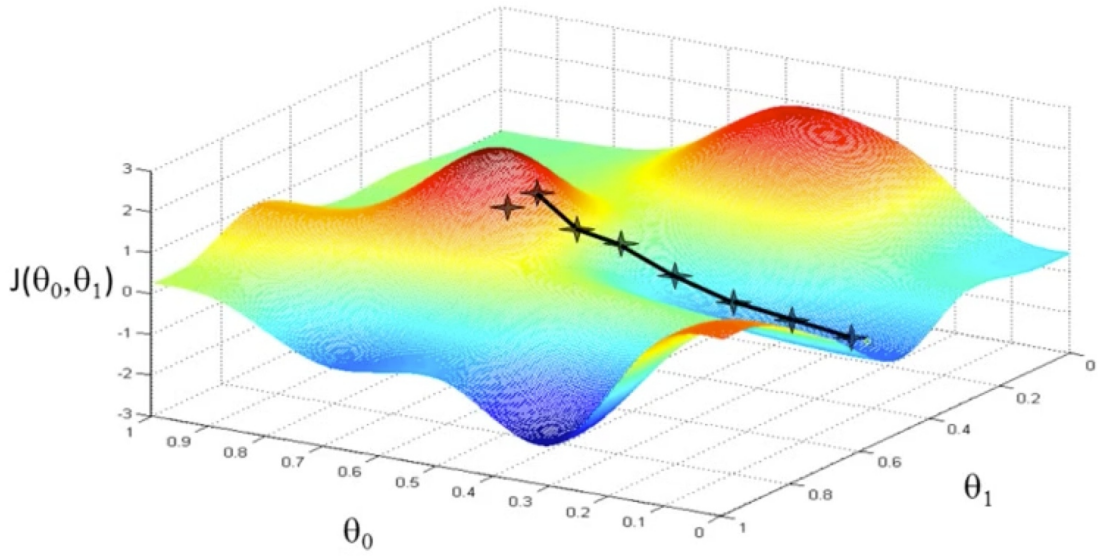
$$h(x) = \theta_0 + \theta_1 * x_i + \theta_2 * x_2 + \ldots + \theta_n * x_n$$

为方便我们使用矩阵来表达，$h(x) = \theta^T * x$，其中$\theta^T$为$\theta$的转置。为求解函数$h(x)$，我们希望找出一组$\theta$，使得$h(x) - y$无限趋近0，此处我们引入梯度下降算法求解问题。

# 3.梯度下降算法

## 3.1梯度下降算法简述

实际生活中我们有时也利用梯度下降算法，比如我们处在一座山的某处位置，但我们并不知道如何下山，于是决定走一步算一步，但每次都沿着最陡峭的地点下山，也就是沿着梯度的负方向前进。但有事也会遇见问题，不能每次都能到达山脚，可能到达山峰的某个局部最低点。




从上面解释可以看出，梯度下降不一定能够找到全局最优解，有可能是局部最优解，但此种方法已能帮助我们求解线性回归问题。另外如果求解的函数是凸函数，梯度下降法得到得解一定是全局最优解。

## 3.2梯度下降算法相关概念

求解梯度下降算法之前，我们先了解相关概念。

- **步长（Learning Rate）**：步长决定梯度下降算法过程中，每步沿梯度负方向前进的长度。
- **特征（Feature）**：即上述描述的$x_i, y_i$
- **假设函数（Hypothesis Function）**：监督学习中，为了拟合输入样本，而使用假设函数$h(x)$
- **损失函数（Loss Function）**：为了评估模型拟合的好坏，通常用损失函数来度量拟合的程度。损失函数越小，意味着拟合的程度越好，对应的模型参数即为最优参数。线性回归损失函数为

$$J(\theta) = \frac{1}{2m} * \sum_{i=1}^{n} (h(x)^{(i)} - y^{(i)})$$

我们利用梯度下降算法，目标便是找到一组$\theta$使得$J(\theta)$达到最小。

## 3.3梯度下降算法过程

- 随机选取一组$\theta$。

- 不断变化$\theta$，让$J(\theta)$变小。$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)\ j=0,1,2...n$，$\theta_j$是n+1个值同时变化。$\alpha$表示学习速率，目标求最小值，因此沿负梯度方向下降，故$\theta$前为负号。$\alpha\frac{\partial}{\partial\theta_j}$是对$J(\theta)$的偏导。
- 直到$J(\theta)$得到最小值。

$\alpha\frac{\partial}{\partial\theta_j}$是对$J(\theta)$的偏导求解过程如下：$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{\partial}{\partial\theta_j}\frac{1}{2m}(h_\theta(x)-y)^2$
$= 2\cdot\frac{1}{2}(h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x)-y) = (h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x)-y) = (h_\theta(x)-y)\cdot x_j$ 因此梯度下降算法的最终表述为

Repeat Until Convergence{ $\theta_j := \theta_j - \alpha\sum_{i=1}^{n}((h_\theta(x^{(i)})-y^{(i)})\cdot x_j)$

for every $j$

}

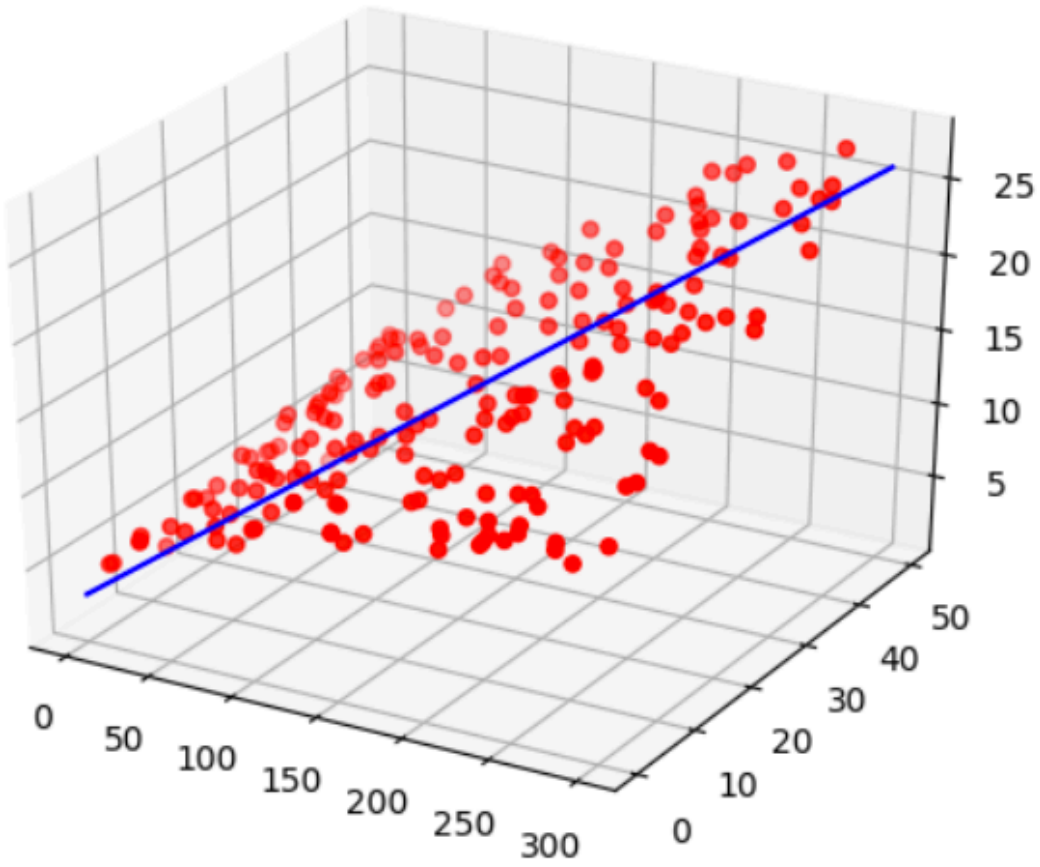
梯度下降算法需多次迭代、算法复杂度为$O(kn^2)$。当利用梯度下降算法求得一组$\theta$时我们便能得到线性回归函数。

## 4.线性回归算法实现

为研究公司盈利提升幅度受电视、广播、报纸的投入的影响程度，利用多元线性回归来分析数据。其中数据下载地址在这儿，为能够绘制出三维图片，此处只选择电视、广播的广告投入对公司盈利提升幅度的影响。

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
from mpl_toolkits.mplot3d import axes3d
import seaborn as sns

#read_csv
readdata=pd.read_csv('data/Advertising.csv')
data=np.array(readdata.values)

#训练数据
X_train=data[0:150,1:3]
Y_train=data[0:150,3]

#测试数据
X_test=data[150:200,1:3]
Y_test=data[150:200,3]

#回归分析
regr = linear_model.LinearRegression()
#进行training set和test set的fit，即是训练的过程
regr.fit(X_train, Y_train)

# 打印出相关系数和截距等信息
print('Coefficients: \n', regr.coef_)
print('Intercept: ', regr.intercept_)
```

```
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((regr.predict(X_test) - Y_test) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(X_test, Y_test))

#得出回归函数 并自定义数据
X_line=np.linspace(0,300)
Y_line=np.linspace(0,50)
Z_line=0.04699836*X_line+0.17913965*Y_line+3.00431061176

#画图
fig=plt.figure()
ax = plt.subplot(111, projection='3d')  # 创建一个三维的绘图工程
ax.scatter(data[:,1],data[:,2],data[:,3],c='red',)  # 绘制数据点
ax.plot(X_line,Y_line,Z_line,c='blue')#绘制回归曲线
plt.show()
```




其中红色为数据点，蓝色线便为我们回归之后的曲线，这里我们是利用sklearn进行线性回归分析，后续会写出sklearn教程。如有错误之处还请指正，谢谢。

## 5.推广

更多内容请关注公众号'谓之小一'，若有疑问可在公众号后台提问，随时回答，欢迎关注，内容转载请注明出处。

「谓之小一」希望提供给读者别处看不到的内容，关于互联网、数据挖掘、机器学习、书籍、生活……

- 知乎：@谓之小一
- 公众号：@谓之小一
- GitHub：@weizhixiaoyi
- 技术博客：https://weizhixiaoyi.com


长按关注微信公众号