实战案例1：梯度下降算法的实现

作者： Robin 日期： 2018/09 数据集来源： scikit-learn dataset 声明： 小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利

1. 案例描述

该案例根据生理指数和疾病发展的定量测量值(Y)使用梯度下降算法拟合出一条直线。 糖尿病数据集 糖尿病数据集包含442个病人测量的10个生理学变量 (年龄、性别、体重、血压)，以及一个一年后病情发展的标记。

2. 数据集描述

- scikit-learn提供的糖尿病数据集，包括生理指数数据文件和标签文件。
- 数据字典
  - **Age**: 年龄，浮点型
  - **Sex**: 性别，浮点型
  - **Body mass index (BMI)**: 体重指数，浮点型
  - **Average blood pressure**: 平均血压，浮点型
  - **S1**: 血清测量值1，浮点型
  - **S2**: 血清测量值2，浮点型
  - **S3**: 血清测量值3，浮点型
  - **S4**: 血清测量值4，浮点型
  - **S5**: 血清测量值5，浮点型
  - **S6**: 血清测量值6，浮点型
  - **Y**: 一年后病情发展的标记，整型

3. 任务描述

- 利用NumPy实现梯度下降算法，并检测哪个生理指数和病情发展呈较强的线性关系。

4. 主要代码解释

- 代码结构

```
lect01_proj
├── data
    ├── diabetes_data.csv    # 生理指数数据文件
    ├── diabetes_target.csv    # 标签文件
├── main.py      # 主程序
├── lect01_proj_readme.pdf  # 案例讲解文档
```

- **main.py**

为了使用梯度下降算法中同一形式的导数公式，需要为每条数据的x添加`1`，即为x向量添加一列全1的向量。

```
def main():
    ...
    # 添加一列全1的向量
    x = np.hstack((np.ones_like(x), x))
    ...
```

- **main.py**

`gradient_descent()`函数中的每次迭代需要获取损失值`cost`和参数的梯度值`grad`。注意这里的学习率`alpha`要根据数据进行选择，如果过大，会造成无法完成参数的优化；如果过小，优化时间会增加。

```
def gradient_descent(x, y, max_iter=1500, alpha=0.01):
    ...
    grad, cost = get_gradient(theta, x, y)
    ...
```

- **main.py**

`get_gradient()`函数通过NumPy实现了向量化的求导公式用于计算梯度，同时通过向量化的方式计算了损失值。

```
def get_gradient(theta, x, y):
    ...
    grad = 1.0 / m * error.dot(x)
    cost = 1.0 / (2 * m) * np.sum(error ** 2)
    ...
```

5. 案例总结

- 该项目通过使用NumPy完成梯度下降算法完成对数据的拟合，巩固并应用了以下知识点:
    - 向量化的思维方式
    - NumPy的使用
    - 梯度下降算法的原理

6. 课后练习

- 尝试不同的学习率观察对优化结果的影响

参考资料

1. 梯度下降法
2. 随机梯度下降
3. NumPy快速入门
4. NumPy教程