

## 一、Numpy 基本操作（提交所有代码截图及运行结果）

### 1、导入 numpy 库

### 2、建立一个一维数组 a，初始化为[4,5,6]

(1) 输出 a 的类型（type）

(2) 输出 a 的各维度大小（shape）

(3) 输出 a 的第一个元素

### 3、建立一个二维数组 b，初始化为[[4,5,6],[1,2,3]]

(1) 输出 b 的各维度大小（shape）

(2) 输出b[0,0],b[0,1],b[1,1]这三个元素

### 4、建立矩阵

(1) 建立一个大小为 $3 \times 3$ 的全 0 矩阵 c

(2) 建立一个大小为 $4 \times 5$ 的全 1 矩阵 d

(3) 建立一个大小为 $4 \times 4$ 的单位矩阵 e

### 5、建立一个数组 f，初始化为[0,1,2,3,4,5,6,7,8,9,10,11]（arange）

(1) 输出 f 以及 f 的各维度大小

(2) 将 f 的 shape 改为 $3 \times 4$ （reshape）

(3) 输出 f 以及 f 的各维度大小

(4) 输出 f 第二行（f[1,:]）

(5) 输出 f 最后两列（f[1:3,:]）

(6) 输出 f 第三行最后一个元素（使用-1 表示最后一个元素）

## 二、Tensorflow 练习（提交每个练习的实现步骤描述以及下面要求提交的结果）

### 1、线性回归

(1) 生成训练数据，数据为带有服从-0.5 到 0.5 的均匀分布噪声的正弦函数，代码如下：

```
num_observations = 100
x = np.linspace(-3, 3, num_observations)
y = np.sin(x) + np.random.uniform(-0.5, 0.5, num_observations)
```

画出这 100 个样本的散点图。（提交散点图）

(2) 使用 tensorflow 实现线性回归模型，训练参数w和b。

$$y = w * x + b$$

(3) 输出参数 $w$ 、 $b$ 和损失。(提交运行结果)

(4) 画出预测回归曲线以及训练数据散点图，对比回归曲线和散点图并分析原因。

(提交图片及分析)

2、线性回归（使用多项式函数对原始数据进行变换）

(1) 生成训练数据，数据同上

(2) 使用 tensorflow 实现线性回归模型，这里我们假设 $y$ 是 $x$ 的 3 次多项式，那么我们可以将数据扩展为： $x$ 、 $x^2$ 、 $x^3$ 三维数据，此时模型变为：

$$y = w_1 * x + w_2 * x^2 + w_3 * x^3 + b$$

(3) 训练模型并输出参数 $w_1$ 、 $w_2$ 、 $w_3$ 、 $b$ 和损失。(提交运行结果)

(4) 画出预测回归曲线以及训练数据散点图，对比回归曲线和散点图并分析原因。

(提交图片及分析)

3、Softmax 分类

(1) 获取 MNIST 数据集，每张图片像素为 $28 \times 28$

(2) 模型架构为：

$$y = \text{softmax}(w * x + b)$$

其中 $w$ 的维度为 $784 \times 10$ ， $b$ 的维度为10。

(3) 画出训练和测试过程的准确率随迭代次数变化图，画出训练和测试过程的损失随迭代次数变化图。(提交最终分类精度、分类损失以及两张变化图)