线性回归与分类实验

一、简介

在本次实验中，我们提供了两份数据：

1） 一个是 Emoji 的数据，是在课程 PPT 中展示的数据，如图 1 所示，这份数据是用来进行线性感知机实验的。

图 1、Emoji 数据集

2） 一个是用来进行年龄估计的数据，如图 2 所示，这个数据集给定一个图像，来估计图像中的人的年龄。我们已经采用神经网络从图像中抽取了一个 2048 维的特征，你所需要做的是从 2048 维的特征来估计人的年龄。

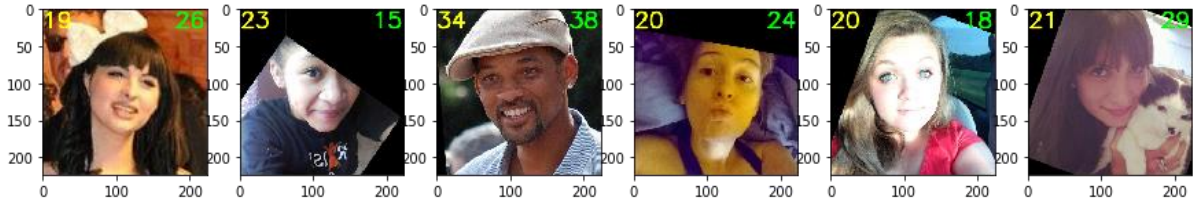

图 2、年龄估计数据集

在本次实验中，你需要完成两部分的内容；

1） 需要完成 LinearRegression-publish.ipynb 中如下三个函数缺失的部分。

a) def closed_form_solution(age, features)

b) def gradient_descent(age, feature)

c) def stochastic_gradient_descent(age, feature)

2）需要完成 LinearPerceptron-publish.ipynb 中

class PrimalPerceptron(object)这个类缺失的部分。

二、线性回归

在线性回归实验中，我们将数据划分成训练集（training），验证集（validation）和测试集（testing）三个部分。我们通过函数 prepare_data 来提供着三部分数据。

- 训练集数据包含一个3995×2048的矩阵X_{train}用来记录输入的特征，用一个3995的向量$\mathbf{y}_{train}$用来记录对应的年龄。
- 验证集数据包含一个1500×2048的矩阵X_{val}用来记录输入的特征，用一个1500的向量$\mathbf{y}_{val}$用来记录对应的年龄。
- 训练集数据包含一个500×2048的矩阵X_{test}用来记录输入的特征，这组数据的实际年龄没有提供，我们在程序中已经提供了每个算法针对测试数据保存的功能，最后会生成 cfs.txt,gd.txt 和 sgd.txt 三个文件，分别对应三个优化算法，**这个结果将作为你得分的重要依据，如果你在训练集上的结果达不到我们的基准值，会被扣分。**

1、注意事项

1）你仅仅能够适用训练集数据来优化权重参数$\mathbf{w}$和偏置b。

2）我们在程序中设置的迭代步骤是不允许被改变的，这是为了保证大家算法的公平性。

3）在 gradient_descent 和 stochastic_gradient_descent 里面，你可以采用验证集来选择最佳参数。

4）我们你的估计结果是否达到基准的依据是你估计的年龄和实际年龄的差异，计算绝对值，然后统计 500 个样本下绝对值的平均值。

2、权重和偏置

为了简洁起见，这里采用X来代表训练的特征，用$\mathbf{y}$来表示相应的年龄，根据刚才我们的数据集描述，X是一个3995×2048的矩阵，$\mathbf{y}$是3995的向量，你的任务是找到一个权重$\mathbf{w}$和偏置b。这里权重$\mathbf{w}$应该是一个2048的向量，偏置b应该是一个标量。根据这个设定，对第i号样本，他的估计年龄应该是：

$$\hat{y}_i = \sum_{j=1}^{2048} X_{ij} w_j + b$$

3、解析解 closed_form_solution

在这个函数里面你需要应用最小二乘法找到权重和偏置的最佳值。这里我们对矩阵和向量的运算做个简单的普及，假设X_i是矩阵X的一行，是个1×2048的行向量，我们把权重$\mathbf{w}$当成是一个2048×1的列向量，偏置b是一个1×1的标量，那么损失函数可以写成：

$$\mathcal{L}(\mathbf{w},b) = \frac{1}{3995}\sum_{i=1}^{3995}(X_i\mathbf{w}+b-y_i)^2$$

这个损失函数相对于权重和偏置的偏导数如下所示：

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial\mathbf{w}} = \dfrac{1}{3995}\displaystyle\sum_{i=1}^{3995}(X_i\mathbf{w}+b-y_i)X_i^T \\ \dfrac{\partial\mathcal{L}}{\partial b} = \dfrac{1}{3995}\displaystyle\sum_{i=1}^{3995}(X_i\mathbf{w}+b-y_i) \end{cases}$$

让这两个偏导数为 0，则可以求出权重和偏置的解析解。

4、梯度下降 gradient_descent

梯度下降算法是通过优化来求解最佳权重和偏置的方法。首先可以通过numpy 随机变量生成器 random 中高斯噪声生成器来生成一个初始的权重和偏置$(\mathbf{w}^0,b^0)$，然后迭代更新。假设目前已经更新了t步，你已经获得的了更新后的参数$(\mathbf{w}^t,b^t)$，这时候你可以将这个更新后的参数代入到上述求解梯度的公式中，获得在当前点的梯度。

$$\begin{cases}\dfrac{\partial \mathcal{L}}{\partial \mathbf{w}}=\dfrac{1}{3995}\sum\limits_{i=1}^{3995}(X_i\mathbf{w}^t+b^t-y_i)X_i^T\\ \dfrac{\partial \mathcal{L}}{\partial b}=\dfrac{1}{3995}\sum\limits_{i=1}^{3995}(X_i\mathbf{w}^t+b^t-y_i)\end{cases}$$

然后根据这个梯度，结合步长α可以更新权重和偏置

$$(\mathbf{w}^{t+1},b^{t+1})=(\mathbf{w}^t,b^t)-\alpha\left(\frac{\partial \mathcal{L}}{\partial \mathbf{w}},\frac{\partial \mathcal{L}}{\partial b}\right)$$

5、随机梯度下降 stochastic_gradient_descent

随机梯度下降方法，每次从全部样本中随机抽取 B 个样本，然后利用这 B 个样本构建一个损失函数，计算梯度，然后更新权重和偏置。除了梯度计算不是相对于全部的 3995 个样本，而是相对于随机抽取的 B 个样本外，其他更新迭代步骤都是一样的。

三、线性分类

在线性分类器这一部分，你需要做三个事情：

1）在__init__这个函数中初始化权重以及偏置，可以通过 numpy 随机变量生成器 random 中高斯噪声生成器来生成一个初始的权重和偏置。

2）填充 predict 函数，这个函数返回两个值，一个是 preds，这个记录着线性感知机在进行 sign 函数运算之前的实数值，另外一个是 y_hat 记录着进行 sign 函数运算之后的结果

3）填充 update 函数，这个函数实现的是梯度下降法，轮询一次所有的样本，更新参数。

如果你所有的步骤都写对了，最后你看到经过若干次迭代之后，你会获得下图的结果。

图 3、Emoji 的正确分类结果