内容简介

在处理完模型输入变量后，本节将开始模型的构建与探讨工作，共涉及到三种模型：MLP 模型、RBF 模型、Lasso 模型。主要包括以下内容：

1. 主要内容

- 关于 MLP 模型和 RBF 模型的相关介绍；
- 搜索案例中探讨的三个模型的最优超参数组合；
- 对三种模型进行比较。

2. 学习目标

学完本节，能解决以下问题：

- 新增的 MLP 模型和 RBF 模型什么原理？如何使用相关的类进行数据分析？
- 为什么使用平均绝对误差评估本次案例的三个模型？

MLP 算法介绍

MLP 模型是一种简单的神经网络。在介绍 MLP 模型之前，我们需要对神经网络、神经元等基础概念进行介绍。

1. 神经网络

神经网络（本处特指人工神经网络）是由具有适应性的简单单元组成的广泛、并行、互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应。

在生物神经网络中，每个神经元与其他神经元相连，当一个神经元“兴奋”时，就会向相连的神经元发送化学物质，从而改变这些神经元内的电位；如果某神经元的电位超过了一个阈值（threshold），它就会被激活，即“兴奋”，并向其它神经元发送化学物质。M-P 神经元模型即是对生物神经元的一种模拟。

2. M-P 神经元

M-P 神经元（McCulloch-Pitts neuron）模型如下图所示。

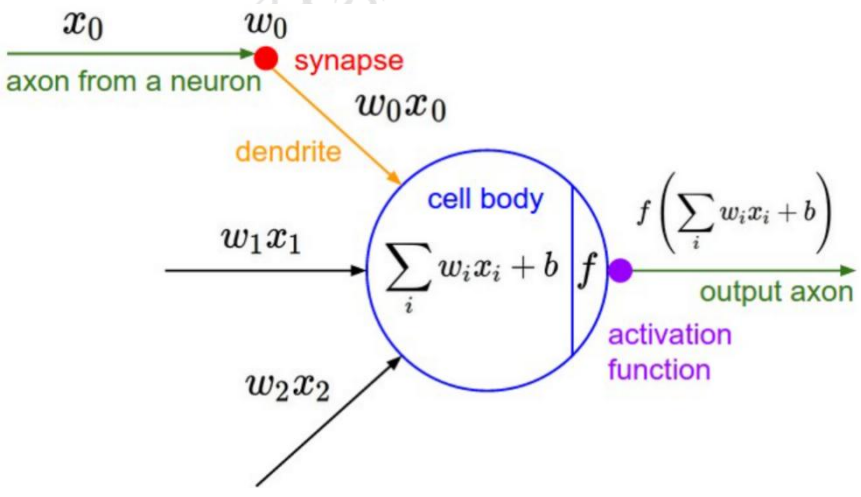


图 1 M-P 神经元（McCulloch-Pitts neuron）

图中的神经元接收到其它神经元传递过来的输入信号（图中包含 3 个信号 x_0, x_1, x_2），信号通过带权重（w_0, w_1, w_2）的连接进行传递。神经元接收到的总输入值将与神经元的阈值（$-b$）进行比较，然后通过**激活函数**（activation function）处理以产生神经元的输出。

从生物学的角度，理想的激活函数是阶跃函数，如下图 2 所示，它将输入值映射为输出值 0/1，0 对应于神经元抑制，1 对应于兴奋。

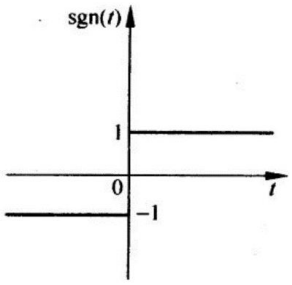


图 2 阶跃函数

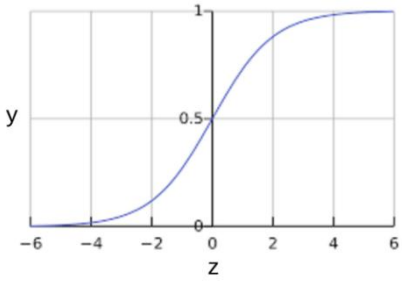


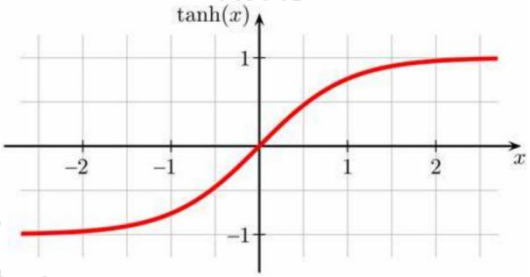
图 3 Sigmoid 函数

但是在实际计算中，阶跃函数具有不连续、不光滑等不太好的性质，因此常用 Sigmoid 函数作为激活函数，如图 3 所示。

除此以外，常用激活函数还有双曲正切函数（tanh），公式为:

$$f(z)=\tanh(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

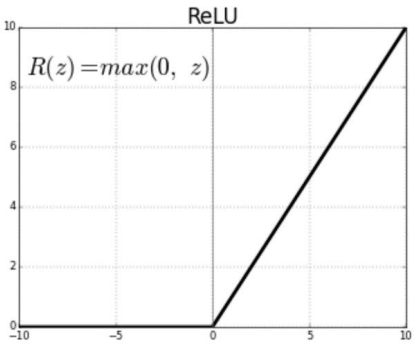
函数图像为：



以及 relu 函数，其公式为：

$$f(x) = \max(0, x)$$

函数图像为：



把许多个这样的神经元按一定的层次结构连接起来，就得到了**神经网络**。

3. 感知机

感知机（Perceptron）是由两层神经元组成，如下图所示。输入层接受外界输入信号后传递给输出层，输出层是 M-P 神经元。

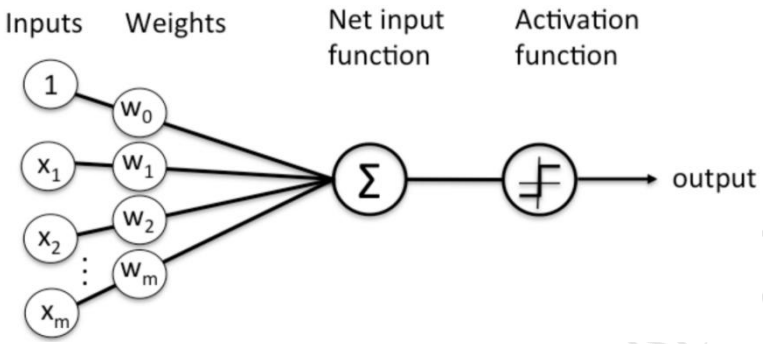


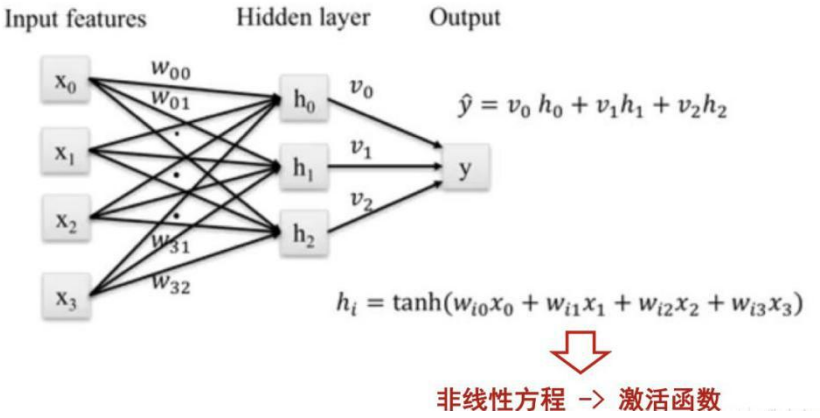
图 4 感知机（Perceptron）

4. 多层感知机

多层感知机（Multi-Layer Perceptron，MLP）是一种神经网络，由输入层（input layer），隐藏层（hidden layers），输出层（output layer）三部分组成。

每层神经元与下一层神经元互连，隐藏层可以有多层，每个隐藏层由多个神经元组成。如下图所示：

其中输入层负责接收样本特征，隐层与输出层神经元对信号进行加工，最终结果由输出层神经元输出。



5. MLP 模型的具体步骤

1）输入特征经过连接节点的权重传入下一层，上一层的输出是下一层的输入。

2）隐层中的每个神经元负责对输入求和，然后根据激活函数转化为输出，也是下一层的输入。

3）MLP 模型的所有参数就是各个层之间的权重以及阈值。

神经网络的学习过程就是求解最佳参数的过程（也称为优化过程）。

其中**误差逆传播**（error BackPropagation, BP）算法是大部分神经网络在训练过程中所采用的算法，BP 算法的基本思路是：

首先随机初始化所有参数；然后在迭代的每一轮中，计算误差目标函数来衡量预测 y 值与实际 y 值的差异；最后基于梯度下降（gradient descent）策略，以目标的负梯度方向对现有参数进行调整，使得预测 y 值尽可能接近实际的 y 值，迭代直到满足某个条件为止（比如误差足够小、迭代次数足够多时）。

最简单的误差目标函数就是**均方误差**（Mean Squared Error, MSE）。

设样本数为 m，则

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

MSE 和最小二乘法中的损失函数计算公式相同。后续为了防止神经网络过拟合，通常还会在误差目标函数中引入正则化项。

注：

1）神经网络的优化过程涉及较为复杂的数学运算和推导，感兴趣的同学可以进一步阅读《机器学习》等书籍；

2）MLP 模型可以通过调用 scikit-learn 的 MLPRegressor 类来实现。

模型评估

在之前介绍线性回归模型的案例中，我们介绍了评价线性回归模型效果的指标 R^2，但是该指标不适用于更广泛的回归模型，如 MLP 和 RBF；对于回归模型来说，也不能使用评价分类模型的准确率指标。这里我们介绍一个新的评价指标：

平均绝对误差（Mean Absolute Error, MAE），设共有 m 个样本，则：

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)|$$

MAE 和与上一个文档中提到的 MSE（均方误差）都是回归模型的测评指标，两者的区别本质上就是 l1 范数和 l2 范数的区别。因为 MSE 计算的是差的平方，因此过大的差值会导致误差增大，所以 MSE 对异常值更加敏感。

在本次案例中，由于转换后的公司概述、岗位概述等字段异常值不太好处理，因此采用 MAE 来评估三个不同的回归模型。

RBF 算法介绍

1. RBF 模型

RBF（Radial Basis Function，径向基函数）网络是一种单隐层神经网络，它使用径向基函数作为隐层神经元激活函数，而输出层则是对隐层神经元输出的线性组合，如下图所示：

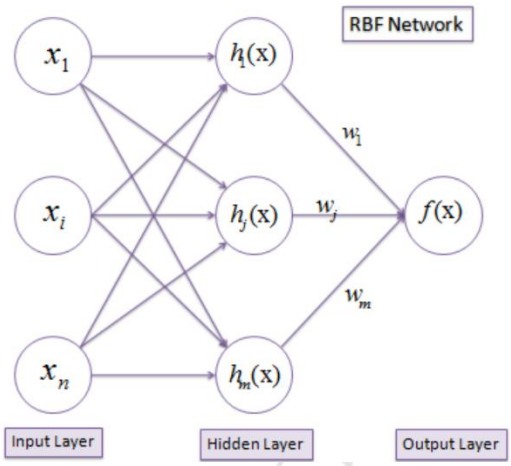


图 1 RBF（Radial Basis Function，径向基函数）模型

隐层中的每个神经元接收经过径向基函数处理过的输入，假定输入为 n 维向量 x，则 RBF 网络的输出可表示为：

$$\Phi(x) = \sum_{i=1}^{q} w_i \rho(x, c_i)$$

其中：q 为隐层神经元个数；

c_i和 w_i分别表示第 i 个隐层神经元所对应的中心和权重；

$\rho(x, c_i)$是径向基函数。

2. 径向基函数

径向基函数通常定义为样本 x 到数据中心 c_i之间欧氏距离的单调函数。

常用的高斯径向基函数公式：

$$\rho(x, c_i) = e^{-\beta_i ||x - c_i||^2}$$

注：目前还没有可直接调用实现的 RBF 模型，因此在代码中人工实现了一个简单版本。

3. 平均绝对误差

在之前介绍线性回归模型的案例中，我们介绍了评价线性回归模型效果的指标 R^2，但是该指标不适用于更广泛的回归模型，如 MLP 和 RBF；也不能使用评价分类模型的准确率指标。这里我们介绍一个新的评价指标：

平均绝对误差（Mean Absolute Error, MAE），设共有 m 个样本，则：

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)|$$

MAE 和与上一个文档中提到的 MSE（均方误差）都是回归模型的测评指标，两者的区别本质上就是 l1 范数和 l2 范数的区别。因为 MSE 计算的是差的平方，因此过大的差值会导致误差增大，所以 MSE 对异常值更加敏感。

在本次案例中，由于转换后的公司概述、岗位概述等字段异常值不太好处理，因此采用 MAE 来评估三个不同的回归模型。

模型特征权重比较

将现有的 6 个特征作为输入，岗位薪资作为输出构建线性回归模型，我们可以通过比较最终模型中特征的权重，探究哪些因素和岗位薪资的相关性较高。

首先通过 *X.shape[1]*可以得到模型的输入变量共有 38 个维度，而最终非零特征只剩余 23 个，非零特征权重如下所示：

```
以下为非零特征及对应系数：
company_financing_stage_x0_B轮  0.2736
company_financing_stage_x0_不需要融资   0.2035
company_financing_stage_x0_已上市       -0.2958
company_people_x0_100-499人     0.5581
company_people_x0_1000-9999人   -0.5832
company_people_x0_10000人以上   -0.9264
job_edu_require_x0_学历：博士   11.5378
job_edu_require_x0_学历：本科   -0.1615
job_edu_require_x0_学历：硕士   1.0539
job_exp_require_x0_经验：1年以内        -0.9442
job_exp_require_x0_经验：3-5年  2.7172
job_exp_require_x0_经验：5-10年 5.4585
job_exp_require_x0_经验：应届生 -5.7815
job_exp_require_x0_经验：经验不限       -0.0387
company_5_x0_0  -4.2240
company_5_x0_1  -1.0967
company_5_x0_3  1.8506
company_5_x0_4  5.2928
job_5_x0_0      -13.5623
job_5_x0_1      -6.8501
job_5_x0_2      -1.5246
job_5_x0_3      2.8802
job_5_x0_4      8.4057
```

从整体上来看，公司融资情况、公司员工数量和岗位薪资的相关性较弱，说明针对机器学习这一类的岗位，公司规模与岗位薪资无明显相关性。

岗位学历要求中，学历要求越高，和岗位薪资的正相关性越高，其中博士的相关性最高。

在工作经验要求中，应届生、1 年以内或经验不限有较弱的负相关，3 年以上具有一定的正相关，说明随着工作经验的积累，岗位薪资成正比，这与实际情况比较吻合。

除此之外，转换后的公司概述、岗位概述字段也和岗位薪资有明显的相关性。随着转换后概述的取值越大，正相关性也越大，这也是合理的，因为在构建转换模型的时候，转换结果表示的就是基于概述对岗位薪资等级的预测。由此可知，公司概述、岗位概述

的文本中包含了和岗位薪资很相关的信息，后续可以考虑进一步通过文本分析，找出概述中更具体的与岗位薪资相关的因素。

数据建模与可视化总结

1、新增的 MLP 模型和 RBF 模型什么原理？如何使用相关的类进行数据分析？

（1）MLP 模型

多层感知机（MLP）是一种神经网络，每层神经元与下一层神经元互连，隐藏层可以有多层，每个隐藏层由多个神经元组成。

MLP 模型由输入层，隐藏层，输出层三部分组成。

其中输入特征经过连接节点的权重传入下一层，上一层的输出是下一层的输入；隐层中的每个神经元负责对输入求和，然后根据激活函数转化为输出，也是下一层的输入；同时 MLP 模型的所有参数就是各个层之间的权重以及阈值。

（2）RBF 模型

RBF（径向基函数）网络是一种单隐层神经网络，它使用径向基函数作为隐层神经元激活函数，而输出层则是对隐层神经元输出的线性组合。

其中径向基函数通常定义为样本 x 到数据中心 c_i 之间欧氏距离的单调函数，径向基函数有很多种，其中常用的高斯径向基函数公式：

$$\rho(x, c_i) = e^{-\beta_i||x-c_i||^2}$$

（3）MLP 模型代码实现

MLP 模型通过调用 scikit-learn 的 MLPRegressor 类来实现，主要涉及该函数的五个参数：

hidden_layer_sizes：隐藏层节点数列表，如：[10,10,10]表示有三层隐藏层，每层神经元有 10 个；

activation：激活函数，可以取值'logistic','tanh'和'relu'；

solver: 权重优化算法，可以取值'lbfgs','sgd','adam'，'lbfgs'是一种基于拟牛顿法的优化算法，'sgd'和'adam'是基于随机梯度下降法，在较大的数据集上，adam 的优化效果较好，对于较小的数据集，lbfgs 优化算法可能会有更快的收敛速度与更好的效果；

以及参数 *random_state* 和最大迭代次数 *max_iter*。

然后通过 GridSearchCV 交叉验证直接得到最优超参数组合，具体代码如图所示：

```
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import GridSearchCV

# 创建MLP模型
mod = MLPRegressor(random_state=0, max_iter=8000)
# 设置参数字典
param_dict = {
    'hidden_layer_sizes':[[1], [10], [100], [10,10], [10,10,10], [100,100]],
    'activation':['logistic','tanh','relu'],
    'solver':['lbfgs','sgd','adam']}
# 设置网格搜索参数
grid_search = GridSearchCV(mod, param_grid=param_dict,
                           scoring='neg_mean_absolute_error', cv=4)
# 使用网格搜索模型对训练集进行拟合
grid_search.fit(X_train, y_train)
# 交叉验证最高分
scr = grid_search.best_score_
# 最优超参数
param = grid_search.best_params_
print("最低MAE: %.4f" % abs(scr))
print("最好模型参数设置:", param)
```

```
最低MAE: 4.4328
最好模型参数设置: {'activation': 'logistic', 'hidden_layer_sizes': [100], 'solver': 'adam'}
```

其中因为 GridSearchCV 返回最优超参数的依据是分值最大，而平均绝对误差是要求越小越好，因此设置 scoring 取值为'neg_mean_absolute_error'，为负的平均绝对误差。

（3）RBF 模型代码实现

由于目前还没有可直接调用的 RBF 模型，该模型代码实现会在作业中直接给出，只要求会掌握使用即可。

该模型我们一共设置了四个参数：self-自变量、indim-输入维度、numNotes-隐层节点数目、outdim-输出维度。

在 X，y 已经准备好的前提下，只需要网格搜索隐层节点数目这一个参数，这个数目一

般会大于样本数，小于 X 维度数。代码如下：

```
from sklearn.metrics import mean_absolute_error

# 拆分出验证集
X_t, X_v, y_t, y_v = train_test_split(X_train, y_train,
                                      shuffle=True, random_state=0)
# 网格搜索超参数
for n in [500, 1000, 1500, 2000, 2500]:
    # 创建RBF模型
    rbf = RBF(X.shape[1], n, 1)
    # 训练模型
    rbf.train(X_t, y_t)
    # 使用模型预测
    y_v_pred = rbf.test(X_v)
    # 评估模型
    mae = mean_absolute_error(y_v, y_v_pred)

    print('隐藏层节点数:%d, MAE:%.4f' % (n, mae))
```

```
隐藏层节点数:500, MAE:18.0059
隐藏层节点数:1000, MAE:15.1483
隐藏层节点数:1500, MAE:13.6012
隐藏层节点数:2000, MAE:12.4139
隐藏层节点数:2500, MAE:12.4755
```

最终结果是在隐藏层节点数为 2000 的时候，平均绝对值误差最小。

2、为什么使用平均绝对误差评估本次案例的三个模型？

之前使用的评价线性回归模型效果的指标 R2 不适用于更广泛的回归模型，如 MLP 和 RBF，对于回归模型来说也不能使用评价分类模型的准确率指标，所以引入了两个误差目标函数平均绝对误差（MAE）和均方误差（MSE），其公式分别为：

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)|$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

MSE 计算的是差的平方，过大的差值会导致误差增大，所以 MSE 对异常值更加敏感。而本案例中转换后的公司概述、岗位概述等字段异常值不太好处理，因此采用 MAE 来评估三个不同的回归模型。