医疗花费预测

- 邓新宇
- 18231045
- 北京航空航天大学 高等理工学院
- git仓库地址https://github.com/SnowPhoenix0105/COST.git

一、题目

> Welcome!
>
> 本科机器学习作业2：医疗花费预测
>
> 你的任务是根据数据集中提供的年龄、性别等几项信息，预测个人医疗花费。 预测结果以submission.csv格式压缩成submission.zip文件后再提交。
>
> 注意:
>
> 1. 每个人除打榜之外，最后还需要提交项目技术报告和重要代码。
> 2. 提交格式严格按照数据集中包含的test_sample.csv，请预测最后一列的charges。
>
> 特别注意不要自己变更命名顺序，将对应预测结果替换charges内容即可。

二、数据集分析

通过命令

```
python -m core.preprocessor.detector
```

利用写好的模块对数据集进行检查，得到如下结果:

```
PS D:\Projects\Python\ML_lession_Cost> python -m core.preprocessor.detectors
[2021-01-07 20:26:31]   train-set:
[2021-01-07 20:26:32]   @detect_value_set:      total_num=1070
[2021-01-07 20:26:32]   @detect_value_set:      ages[47]=
[2021-01-07 20:26:32]   @detect_value_set:      sexs[2]={'female', 'male'}
[2021-01-07 20:26:32]   @detect_value_set:      bmis[497]=
[2021-01-07 20:26:32]   @detect_value_set:      childrens[6]={'4', '5', '0', '3', '1', '2'}
[2021-01-07 20:26:32]   @detect_value_set:      smokers[2]={'no', 'yes'}
[2021-01-07 20:26:32]   @detect_value_set:      regions[4]={'northwest', 'southwest', 'northeast', 'southeast'}
[2021-01-07 20:26:32]   @detect_value_set:      chargess[1069]=
[2021-01-07 20:26:32]   test-set:
[2021-01-07 20:26:32]   @detect_value_set:      total_num=268
[2021-01-07 20:26:32]   @detect_value_set:      ages[46]=
[2021-01-07 20:26:32]   @detect_value_set:      sexs[2]={'female', 'male'}
[2021-01-07 20:26:32]   @detect_value_set:      bmis[210]=
[2021-01-07 20:26:32]   @detect_value_set:      childrens[6]={'4', '5', '0', '3', '1', '2'}
[2021-01-07 20:26:32]   @detect_value_set:      smokers[2]={'no', 'yes'}
[2021-01-07 20:26:32]   @detect_value_set:      regions[4]={'southwest', 'northeast', 'northwest', 'southeast'}
[2021-01-07 20:26:32]   @detect_value_set:      chargess[1]={'0.0'}
```

（如果某个属性的值较多，就不进行打印了）

数据一共有6个输入属性和1个输出属性：

属性名	方向	类型	含义
age	in	整数	年龄
sex	in	2类枚举	性别
bmi	in	浮点数	BMI数值
children	in	整数	孩子数量
smoker	in	2类枚举	是否吸烟
region	in	4类枚举	地区
charges	out	浮点数	预计医疗花费

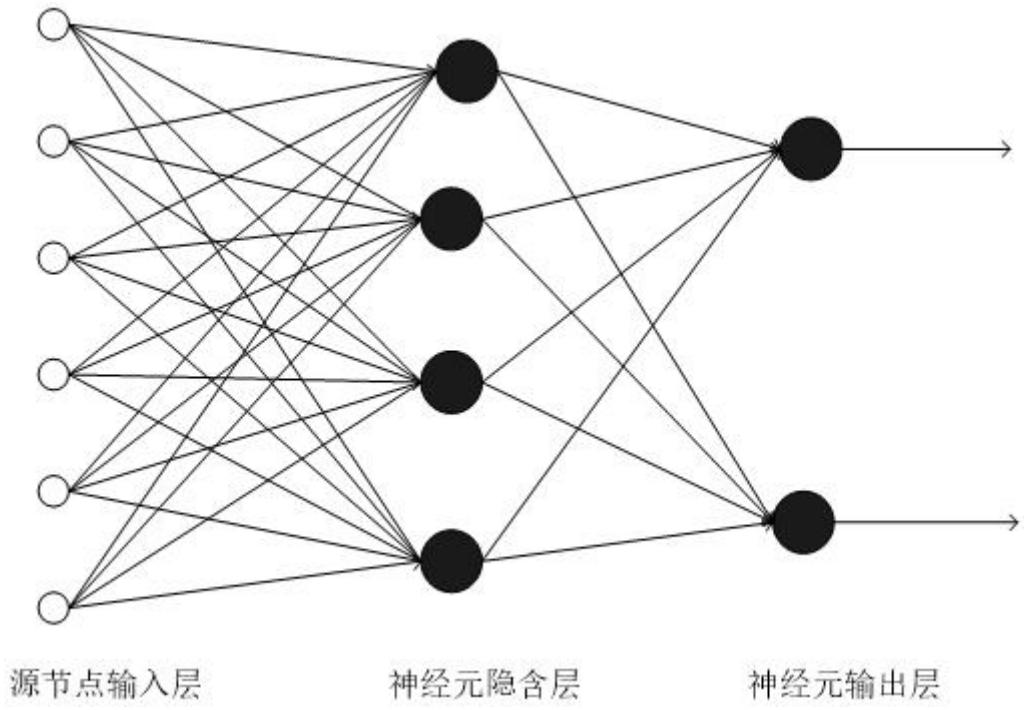
预期结果是一个浮点数，所以本问题是一个回归问题。

将输入数据向量化时，值得注意的是，`region`属性是一个4类的枚举数据，并且四个枚举值分别代表了东北、东南、西北、西南四个方向，但是并不具有次序特征，所以将该属性向量化的时候，应当采取独热码方式，而不能简单将其映射到$[0,1]$区间。因此输入的部分向量化之后应当是一个1维向量，有9个分量。

三、模型

选用向前神经网络进行回归任务训练。

神经网络的大致结构如下：



激励函数备选项有`sigmoid`、`ReLU`，即$z=\frac{1}{1+e^{-y}}$、$z=max\{0,y\}$，后来发现`sigmoid`表现更佳，所以后面都以`sigmoid`作为激励函数来举例。

3.1 正向传播

对于正向传播过程，引入如下记号：

- 记神经网络总层数为M（即有M-1个隐藏层）；
- 记第i层输入、输出维数为N_{in}^i、N_{out}^i，其中$i=1,2,\dots,M$;
- 记第i层第j/k个神经元的输入、净输入、输出分别为x_j^i、y_k^i、z_k^i，其中$i=1,2,\dots,M$，$j=1,2,\dots,N_{in}^i$，对于$k=1,2,\dots,N_{out}^i$；

它们有如下关系：

$$N_{in}^i=N_{out}^{i-1},\forall i=2,3...M$$

$$x_j^i=z_j^{i-1},\forall i=1,2,\dots,M,\forall j=1,2,\dots N_{in}^i$$

$$y_k^i=\sum_{j=1}^{N_{in}^i}w_{kj}^ix_j^i+b_k^i,\forall i=1,2,\dots,M,\forall k=1,2,\dots,N_{out}^i$$

$$z_k^i=sigmoid(y_k^i),\forall i=1,2,\dots,M,\forall k=1,2,\dots N_{out}^i$$

其中w_{jk}^i、b_k^i需要通过训练得到，若将b_k^i记为$w_0^ij\cdot x_0^i,x_0^i=1$，则可以将$b_k^i$整合入$w_{kj}^i$。简记为$W_i=\{w_{kj}^i\}_{N_{out}^i\times(N_{in}^i+1)}$,并记$X_i=(1,x_1^i,x_2^i,\dots,x_{N_{in}^i}^i)^T$、$Y_i=(y_1^i,y_2^i,\dots,y_{N_{out}^i}^i)^T$、$Z_i=(z_1^i,z_2^i,\dots,z_{N_{out}^i}^i)^T$，则有：

$$X_{i+1}=Z_i=sigmoid(Y_i)=sigmoid(Wi\cdot X_i),\forall i=1,2,\dots,M-1$$

在本问题中，输入为(经度，维度，标价)，记完成任务为$Z_M=(1,0)$，未完成任务为$Z_M=(0,1)$，则有$N_{in}^1=3$，$N_{out}^M=2$。由于本问题为分类问题，故对于神经网络中最后一层的输出z_1^M、z_2^M，需进行进一步独热化处理：

$$(out_1,out_2)=\vec{F}(z_1^M,z_1^M)=\begin{cases}(1,0), & z_1^M\geq z_2^M\\(0,1), & z_1^M<z_2^M\end{cases}$$

$out_1=1$时表示任务被完成，$out_2=1$表示任务未被完成。

3.2 反向传播

利用随机梯度下降法进行反向传播，在一次迭代中，对每个训练样本都进行一次梯度下降。现引入以下记号：

- 记训练集$D=\{d_1,d_2,\dots,d_n\}$；
- 记正确输出为$T_d=(t_1^d,t_2^d,\dots,t_{N_{out}^M}^d)$，其中$d\in D$;
- 记神经网络对于训练样本d正向转播后的总体误差为$E_d=\frac{1}{2}\cdot\sum_{k=1}^{N_{out}^M}(t_k-z_k^M)^2$，其中$d\in D$;
- 记第i层第j个神经元的$\delta_j^i=-\frac{\partial E_d}{\partial y_k^i}$，并简记$\Delta_i=(\delta_1^i,\delta_2^i,\dots,\delta_{N_{out}^i}^i)^T$，其中$i=1,2,\dots,M$，$k=1,2,\dots,N_{out}^i$；
- 记梯度下降的学习率为η，梯度下降w_{kj}^i的改变量为$dw_{kj}^i=-\eta\frac{\partial E_d}{\partial w_{kj}^i}$，$W_i$的该变量为$dW_i$；

则有：

$$\frac{\partial E_d}{\partial w_{kj}^i}=\frac{\partial E_d}{\partial z_k^i}\cdot\frac{\partial z_k^i}{\partial y_k^i}\cdot\frac{\partial y_k^i}{\partial w_{kj}^i}$$

$$\frac{\partial E_d}{\partial z_k^i}=\begin{cases}\frac{1}{2}\cdot\frac{\partial\sum_{k=1}^{N_{out}^M}(t_k-z_k^M)^2}{\partial z_k^M}=z_k^M-t_k^d, & i=M\\ \sum_{k'=1}^{N_{out}^{i+1}}\frac{\partial E_d}{\partial y_{k'}^{i+1}}\cdot\frac{\partial y_{k'}^{i+1}}{\partial z_k^i}=\sum_{k'=1}^{N_{out}^{i+1}}(-\delta_{k'}^{i+1})\cdot w_{k'k}^{i+1}, & i\neq M\end{cases}$$

$$\frac{\partial z_k^i}{\partial y_k^i}=\frac{\partial sigmoid(y_k^i)}{\partial y_k^i}=z_j^i\cdot(1-z_k^i)$$

$$\frac{\partial y_k^i}{\partial w_{kj}^i}=\frac{\partial\sum_{j'=1}^{N_{out}^i}(w_{kj'}^i x_{j'}^i)}{\partial w_{kj}^i}=x_j^i$$

整理得：

$$\Delta_i=\begin{cases}(T_d-Z_M)*Z_M*(1-Z_M), & i=M\\ Z_i*(1-Z_i)*(W_{i+1}^T\cdot\Delta_{i+1}), & i\neq M\end{cases}$$

$$dW_i=\eta\Delta_i\cdot X_i^T$$

这就是每次迭代时更新的权重的改变量。

3.3 其它细节

- 并不是所有属性的值域都在区间$[0,1]$内，这会导致这些属性的权重变高，为了进行平衡，将所有属性的值进行标准化，即通过线性变换将每一列的均值控制为0，方差控制为1，对期望输出也进行这样的处理，需要注意的是，在进行实际预测的时候，要对结果进行逆变换才能得到预测值；
- 设定一个最大迭代次数和目标误差值，这二者满足其一就结束训练迭代，这样可以一定程度下降低过拟合情况，但是也需要通过梯度下降情况来适当扩大最大迭代次数或者降低目标误差值；
- 训练集和测试集较小，所以可以直接将全部数据读入；
- 更改参数尝试发现只有一层隐藏层，并且隐藏层神经元数量为35时训练效果最佳；