中风预测器技术报告

> 姓名：丁盛为
> 学号：20373921
> 班级：200616

项目结构

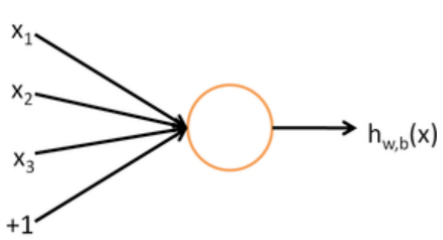
```
|_data
    |_test.csv: 题目给出
    |_testDataSet.csv: 预处理后的test数据集
    |_train.csv: 题目给出
    |_trainDataSet.csv: 预处理后的train数据集
|_src:
    |_main.py: 主程序，实现了训练和预测的主体
    |_MyDataSet.py: 实现了一个继承torch.utils.data.Dataset的类
    |_MyModel.py: 本次作业采用的模型
    |_PreProcess.py: 预处理csv文件的代码
```

模型原理

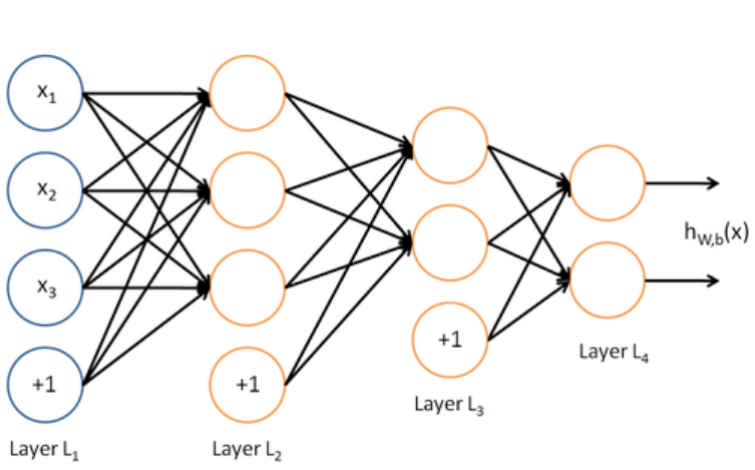
本次作业中采用的是MLP。

直观理解

多层感知器(Multi-Layer Perceptron,MLP)，除了输入输出层，它中间可以有多个隐层，其输入特征值，对特征进行处理后产生输出。神经元的结构如下：



由这样的神经元组成的网络就是MLP：



神经网络的作用，简单来说就是：把神经网络看做一个黑盒，那么$x1$、$x2$、$x3$是这个黑盒的输入X，最右面的$h_{w,b}(x)$是这个黑盒的输出Y。这可以通过一个数学模型来拟合，通过大量训练数据来训练这个模型，之后就可以预估新的样本X 应该得出什么样的 Y。

数学原理

每一个神经元都是在对多个数据进行加权求和，这是在做线性变换，而激活函数则是在做非线性的变换，通过这两者的配合，当网络的神经元数量足够多，层数足够深时，神经网络就拥有了拟合几乎一切分布的能力。因此MLP可用于中风预测。

参数优化

假设W 标识某个神经元对输入信息所做的矩阵计算，b 表示对某个神经元输出结果所加的偏移，则优化的方法为：先初始化一个不靠谱的W 和b，然后用输入x 和W,b 预估$predict_y$，然后根据预估的$predict_y$和实际的$real_y$ 之间的差距来通过梯度下降法更新W,b，然后再继续下一轮迭代，最终逼近正确的W,b

用公式表示反向传播梯度下降法中参数优化的过程，有：$W_k = W_k - \alpha \frac{\partial}{\partial W_k} J(W_{1,2...m}, b_{1,2...m})$，
$b_k = b_k - \alpha \frac{\partial}{\partial b_k} J(W_{1,2...m}, b_{1,2...m})$，其中$\alpha$表示学习率，$J()$表示损失函数。

模型结构

本次作业由于数据量不大，且原始特征的数量也不多，因此使用了两层全连接层组成的MLP来实现中风预测任务。

采用pytorch实现，源码如下：

```
class MyModel(nn.Module):
    def __init__(self):
        super(MyModel, self).__init__()
        self.linear1 = nn.Linear(10, 8)
        self.linear2 = nn.Linear(8, 1)

    def forward(self, input):
        x = F.relu(self.linear1(input))
        x = F.sigmoid(self.linear2(x))
        return x
```

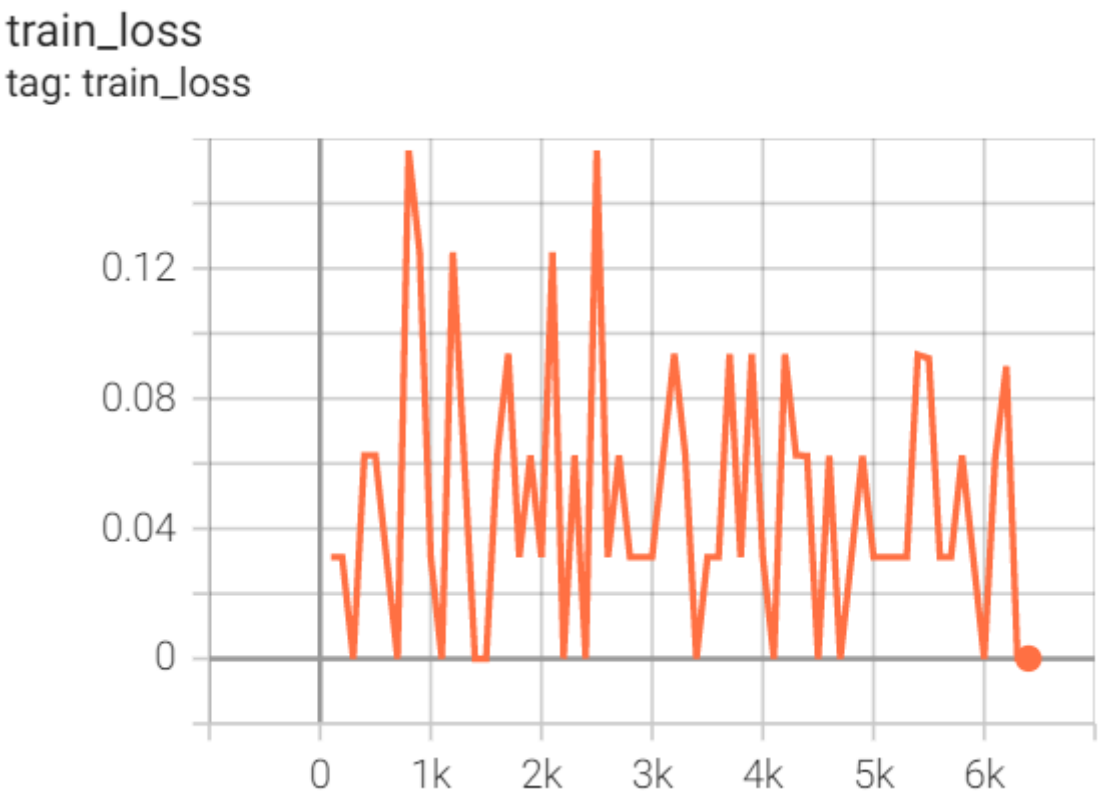
第一层隐层之后所选用的激活函数为常用的relu()，第二层所采用的激活函数则是sigmoid()，因为最后的输出为0或1，因此采用sigmod()将第二个隐层的输出映射到$(0,1)$。当进行预测时，有$predict_y = round(net_output, 0)$。

调参训练过程

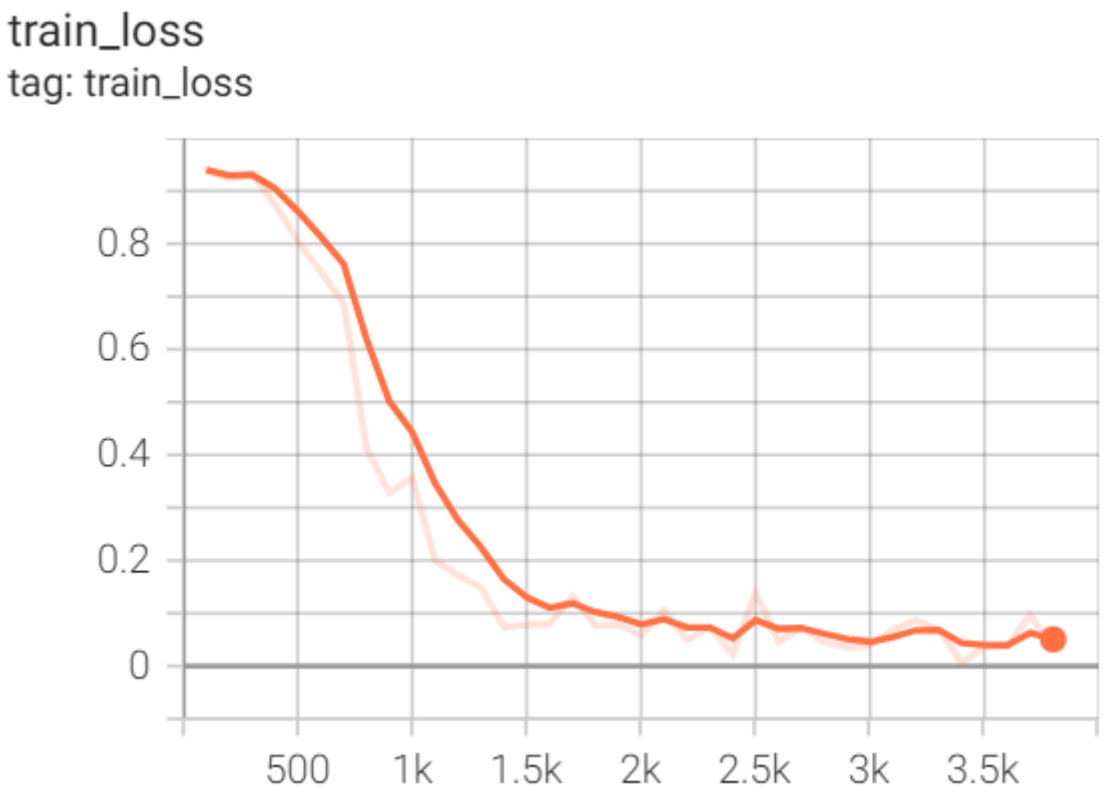
由于本次作业的数据量不大，且我用的是服务器的3090卡，因此初始epoch设置的比较大：设置训练超参数`epochs=50，lr=1e-5，batch_size=32`，本次训练的正确率达到了96%。

12	20373921_丁盛为	1	11/28/22	0.960278 (3)

观察得到loss的曲线：（横轴是train_step）



可见其波动较大，由于网络较小，epochs设的太大了，极有可能出现过拟合的情况，因此减小了epochs=30再次进行训练，



此时的loss曲线下降就显得合理多了，上交后结果与上一次的一致（第二行的为减小epoch之后预测的结果）

#	分数	文件名	提交日期	状态	✔	
1	0.9602780536	submission.zip	11/28/2022 01:10:43	Finished	✔	✚
2	0.9602780536	submission.zip	11/28/2022 15:48:04	Finished		✚

由于此时拟合的效果已经十分不错了，因此停止进一步调整超参数。

数据集预处理

本次作业中的cxv文件中存在一些字符串，需要将他们映射到对应的数字上才可以进行训练，我的映射策略如下：

```
trans_dict = {
    'Female': 0, 'Male': 1, 'Other': 3,

    'No': 0, 'Yes': 1,

    'Self-employed': 0, 'Private': 1, 'Govt_job': 2,
    'children': 3, 'Never_worked': 4,

    'Rural': 0, 'Urban': 1,

    'never smoked': -1, 'Unknown': 0, 'smokes': 1, 'formerly smoked': 2
}
```

同时我舍弃了 `id` 字段，因其仅仅起到一个标记的作用，本身对训练没有影响。

值得一提的是，在bmi字段中存在相应信息的缺省，在文件中为N/A，本次作业中我选择使用0填充的方式对其进行处理。

处理之后的文件为testDataSet.csv和trainDataSet.csv。两者都没有列表头和id，只存有训练所需的数据。处理之后的testDataSet.csv如下所示：

	A	B	C	D	E	F	G	H	I	J
1	0	60	0	0	1	0	0	83.57	24.5	-1
2	0	24	1	0	0	1	1	107.22	35.3	1
3	0	66	0	0	1	2	0	200.49	34.6	1
4	1	44	0	0	1	1	1	99.34	33.1	-1
5	1	73	0	1	1	1	0	62.44	25.2	1

成绩截图

榜上成绩如图：并列第三

#	用户名	登录	上次登录日期	Difference ▲
1	sy2106346	1	11/05/22	1.000000 (1)
2	19376085_文显庆	19	11/15/22	0.961271 (2)
3	20373120_王昶皓	1	11/11/22	0.960278 (3)
4	test_zyy	1	11/05/22	0.960278 (3)
5	20373486_何金威	1	11/09/22	0.960278 (3)
6	19376471_刘骁	2	11/09/22	0.960278 (3)
7	20373360_张铭芳	5	11/17/22	0.960278 (3)
8	20373653_唐怡浜	1	11/26/22	0.960278 (3)
9	20373861_李治圻	8	11/20/22	0.960278 (3)
10	20231231_苏俊行	1	11/27/22	0.960278 (3)
11	20373374_彭卓清	5	11/22/22	0.960278 (3)
12	20373921_丁盛为	2	11/28/22	0.960278 (3)

两次提交结果如图：

#	分数	文件名	提交日期	状态	✔	
1	0.9602780536	submission.zip	11/28/2022 01:10:43	Finished	✔	✚
2	0.9602780536	submission.zip	11/28/2022 15:48:04	Finished		✚