李宏毅深度学习作业一 —— 回归

概要

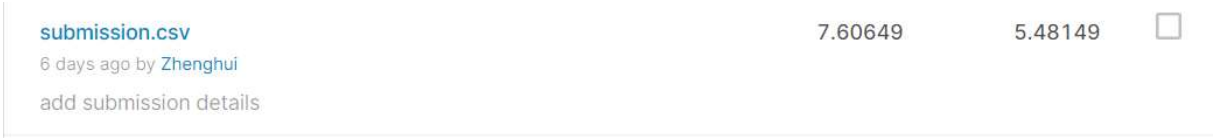
实验介绍：训练数据集给出了 12 个月，每个月前 20 天，每天 24 个小时的 18 个空气污染指数（其中包括 PM2.5 指数）。测试数据集给出 240 组，连续 9 小时的空气污染指数数据，并要求预测第 10 小时的 PM2.5 指数。

实验流程：首先复现了 baseline 模型，提交 Kaggle 达到‘weak baseline’，然后用 sklearn 中的传统机器学习算法拟合数据，最佳模型（Linear Regression）得到的结果与 baseline 几乎一致，其它算法效果无法超越 baseline。接着考虑了特征的相关性，利用关联性较大的特征进行训练，但效果无法超越 baseline（这里我认为与结果相关性不大的特征放入线性模型后，它的权重是趋于 0 的，因此，将相关性不大的数据移除，结果反而可能更差了）。最后我构建了一个 3 层的全连接网络训练数据，得到的测试结果在 Kaggle private leaderboard 排在前 30 位（Strong baseline）。

开源 Baseline 复现

Baseline 提供了有效的数据预处理方式，将训练文件（train.csv）中（4320，24）（4320 = 12*20*18）的数据按月份重新排列，并使得每一行为一组特征向量 X，特征数为 18*9=162。由于每个月只取前 20 天的数据，因此数据不是连续的，只能按照月份分开处理，每个月 20 天，每天 24 小时，一共是 480 个小时的数据，要求用连续 9 小时的数据来预测第 10 小时的 PM2.5，因此每个月可以提取出 480-9+1=472 组特征向量（实际上 baseline 的预处理有问题，它们只提了 471 组特征，不过这无关紧要），按照每个月 471 组处理，一共是 471*12=5652 组训练数据，其中的 80%作为训练集，20%作为验证集。

Baseline 用最基础的线性模型（y = w*x+b）处理特征，用 rmse 作为损失函数，通过反向传播训练出最佳权重，令人惊讶的是这种做法竟然非常有效？在验证集中得出的训练结果已经很准确了（验证集 rmse=5.66，Kaggle private score：7.60，public score：5.48）



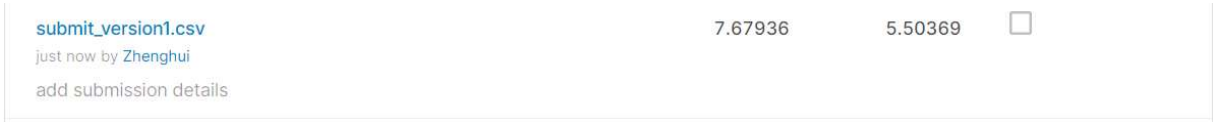
利用 sklearn 库的传统算法进行训练

sklearn 库封装好的模型可以直接拿来使用，训练的很快，因此把能用作回归分析的模型大致都试了一遍（线性回归、决策树、SVM、KNN、Adaboost、GBRT 等），最后发现仍然是线性模型准确率最高（这里实际上挺奇怪的，为啥线性模型会效果最好呢）。

```
In [4]: train_LinearRegression()
        train_DecisionTree()
        train_SVM()
        train_KNN()
        train_Adaboost()
        train_GBRT()
        train_Bagging()
        train_ExtraTree()
```

```
Linear Regression: rmse =  5.668544173979506
TreeDecision Regression: rmse =  9.553007220221396
SVM Regression: rmse =  9.202740382789747
KNN Regression: rmse =  12.07762104680841
Adaboost Regression: rmse =  6.8563457709591535
GBRT Regression: rmse =  6.034201436313543
Bagging regression: rmse =  6.817532139558778
ExtraTree Regression: rmse =  10.482902588708441
```

用表现最好的线性模型拿来测试，验证集 rmse=5.67，Kaggle 上的 Private score：7.68，Public score：5.50，低于原 baseline。



特征分析

162 个特征中，显然其中的 9 个 PM2.5 数据价值很大。其它特征的相关性无法判断。因此我利用预处理后的特征数据贴上标签存入 csv 文件，用 pandas 的 pd.corr()检测每天的 18 个指数与结果的相关性，得到结果如下：

```
Characteristic Correlation:
AMD_TEMP: 0.0013404436338343068
CH4: 0.2614886718884517
CO: 0.3143841434545102
NMHC: 0.3266357570166916
NO: 0.07922619032631435
NO2: 0.47496992755777545
NOx: 0.41147741180872144
O3: 0.37733082179609506
PM10: 0.7567723912073928
PM2.5: 0.9144170870841293
RAIN: -0.07256164379265834
RH: -0.30606952727533404
SO2: 0.4033830975544701
THC: 0.37470174628261627
WD_HR: 0.21344681917760985
WIND_DIR: 0.18693070615207474
WIND_SPEED: -0.07131088008681327
WS_HR: -0.045687287192624486
Out[4]: '\n这里检测出了几个关联性不大的特征（考虑将相关系数小于0.2的舍去），另有两个相关性非常大的特征（>0.7）\n'
```

根据相关性检测结果，PM2.5 与结果相关性达到 90%以上，PM10 达到 76%，其它的指数相对较低，部分指数几乎没有任何关联。

由此，我只保留 PM2.5 数据进行了测试，得到的参数在验证集的 rmse=5.86，低于 baseline 的 5.66：

```
Linear Regression(pm25 data): rmse =  5.861093589266862
```

我又保留 PM2.5 和 PM10 的数据进行了测试，验证集的 rmse=5.77，仍然低于 baseline：

```
Linear Regression(pm10, pm25 data): rmse =  5.770850986437787
```

最后我综合上述的数据和 baseline 的数据（数据作加权），得到的结果略高于 baseline，在 Kaggle Public Leaderboard：5.43，Private Leaderboard：7.52。

神经网络训练

162 维数据训练

3 层神经网络：input_size=162, hidden1_size=64, hidden2_size=16, output_size=1，loss 损失函数采用 mse。

如果使用 162 维的数据，我可以很轻松的让训练数据拟合的非常好（mse 接近 2），但在验证集的表现很不理想。

调参后 mse 在 36 附近（rmse≈6），感觉不是特别理想，没有提交结果

以下为验证集 mse 的最佳结果：

```
After 2526 iterations, loss on x_train is: 34.1267
After 2500 iterations, mse on x_val is: 35.7354
After 2627 iterations, loss on x_train is: 35.8677
After 2600 iterations, mse on x_val is: 36.9597
After 2728 iterations, loss on x_train is: 36.1428
After 2700 iterations, mse on x_val is: 36.227
After 2829 iterations, loss on x_train is: 38.1218
After 2800 iterations, mse on x_val is: 37.2137
After 2930 iterations, loss on x_train is: 29.4338
After 2900 iterations, mse on x_val is: 36.5919
y_pred shape:  (?, 1)
(240, 1)
[[  9.133324 ]
 [ 16.905582 ]
 [ 25.445335 ]
 [ 10.112623 ]
 [ 26.424793 ]
 [ 20.96256  ]
 [ 22.733093 ]
 [ 31.27898  ]
 [ 15.728445 ]
 [ 62.44397  ]
```

18 维数据训练

3 层神经网络： input_size=18, hidden1_size=24, hidden2_size=8, output_size=1

调参后 mse 在 32 附近（rmse≈5.66）

```
After 22000 iterations, mse on x_val is: 32.9745
After 23024 iterations, loss on x_train is: 40.4895
After 23000 iterations, mse on x_val is: 32.9653
After 24025 iterations, loss on x_train is: 26.9094
After 24000 iterations, mse on x_val is: 32.5689
After 25026 iterations, loss on x_train is: 33.8569
After 25000 iterations, mse on x_val is: 32.4803
After 26027 iterations, loss on x_train is: 27.6368
After 26000 iterations, mse on x_val is: 32.9513
After 27028 iterations, loss on x_train is: 35.4356
After 27000 iterations, mse on x_val is: 32.9472
After 28029 iterations, loss on x_train is: 37.7942
After 28000 iterations, mse on x_val is: 33.5738
After 29030 iterations, loss on x_train is: 33.8931
After 29000 iterations, mse on x_val is: 32.469
y_pred shape:  (?, 1)
(240, 1)
```

18 维的数据在 Kaggle 上 Public score：5.76，Private score：7.05，其中 Private score 超越了“Strong Baseline“，排在 private Leaderboard 的前 30 位。奇怪的是，结果在 private 排名中表现得比 baseline 好得多，却在 public

排名中差了一点,且神经网络训练出的 rmse 与线性模型的 rmse 几乎完全一致，是否说明了什么问题？

总结

1. 之前做分类任务比较多，对于回归任务不是特别熟悉（四月份参与过一个华为云大数据挑战赛，与此题目非常类似，是关于道路交通流量预测的，当时直接用的 sklearn 的模型，但没用过神经网络）。而相较于传统的分类任务，我发现只要将神经网络的输出由 n 分类任务的 n 个调整为 1 个，并改变 loss 函数为 mse 函数，就将用于分类的网络转化成了做回归任务的网络。不知道这是不是通用做法，但效果尚可，后面遇到类似问题还能深入考虑。

2. 经过几天的研究训练，我感觉这个数据集整体质量存在一些问题（多个不同的模型得到的 rmse 非常统一地收敛到了 5.66）。这可能导致了神经网络的精确度无法降到特别理想的数值。另外，有一个关键特征无法用上，即时间特征，PM2.5 在 24 小时内的变化与时间一定是有关联的，我们在训练集上可以手动添加上这一特征，但问题是我们不清楚预测集提供的前 9 小时特征究竟是一天中的哪个时间范围，这直接导致了不确定性的增加。如果对数据集进行适当补充，数据集的质量以及拟合效果可能会好得多。

3. 之前还有一些想法，考虑过对 18 个特征每个都按时间进行一次拟合，即：利用前 9 小时的 9 个数据预测出第 10 小时的对应数据，再利用预测出来的 17 个数据对 PM2.5 进行预测，将这个预测与单一使用 PM2.5 预测结果加权平均，效果或许不错，可以尝试。