时间2022.10.16-2022.10.23

工作内容：

1. 在上周日组会后进行改进工作。改进了原有的针对提前24小时需求的预测方式，将输入特征由前十四天的24小时负载数据改为前十四天对应时刻的电力负载数据（即输入一个[1, 14]的特征矩阵）。
   1. 改进前后的预测整体效果（选取测试集中的200个小时）

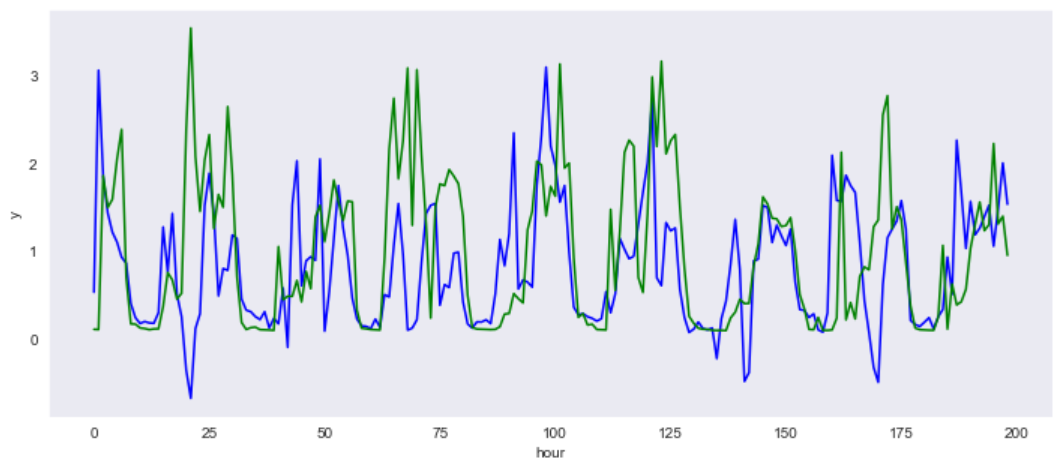
（采用MLP模型，hidden\_layer=1，out\_feature=100）

改进前的预测效果（输入[1, 336]，输出[1, 1]）

（蓝色曲线为预测数值，绿色曲线为实际数值）

RMSE = 0.5735

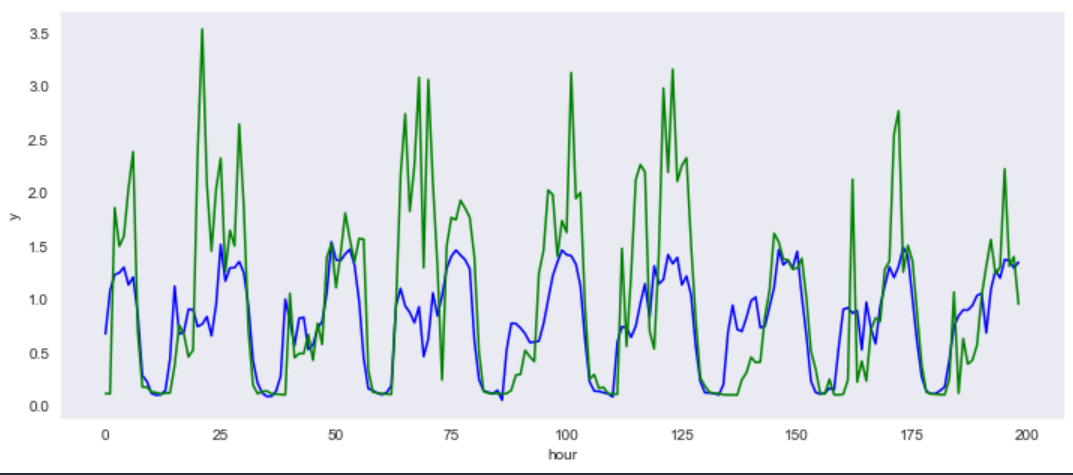
NRMSE = 0.6407



改进后的预测效果（输入[1, 14]，输出[1, 1]）

RMSE = 0.4566

NRMSE = 0.5164



* 1. 效果分析

对改进前后的效果进行分析，发现改进后在整体趋势上提升了预测的准确度，RMSE和NRMSE都较为明显的体现出预测准确度的提升，但是仔细对比会发现改进前对于电力负载的峰值预测效果明显优于改进后对于峰值预测的效果。峰值预测的准确性会极大的影响发电端的控制策略，进而影响到发电端的经济效益。因此对于预测而言不仅要求整体趋势的预测准确性，也要求峰值的准确预测

同时，两种预测方式在变化趋势的预测上，都存在有一定的滞后性，也就是预测出的曲线变化趋势相对于实际曲线的变化趋势存在一定的滞后。

* 1. 对于问题的猜想

这里对提出的第一个问题进行分析。分析改进前后输入的特征，改进前输入的特征是完整的十四天的每一个时刻的电力负载，能完整的体现每天的电力负载的一个整体水平，但是较多的特征输入（输入矩阵为[1, 336]）就弱化了一天中不同时刻在预测中的影响；对于而改进后的输入特征（[1, 14]）而言，由于挑选输入的为14天中同一时刻的电力负载预测数据，因此强化了一天中不同时刻对于电力负载的影响，但很大程度上弱化了负载数据的整体水平。故，改进前对于峰值的预测效果要比改进后的效果好，改进后的预测在整体趋势性上具有明显优势。

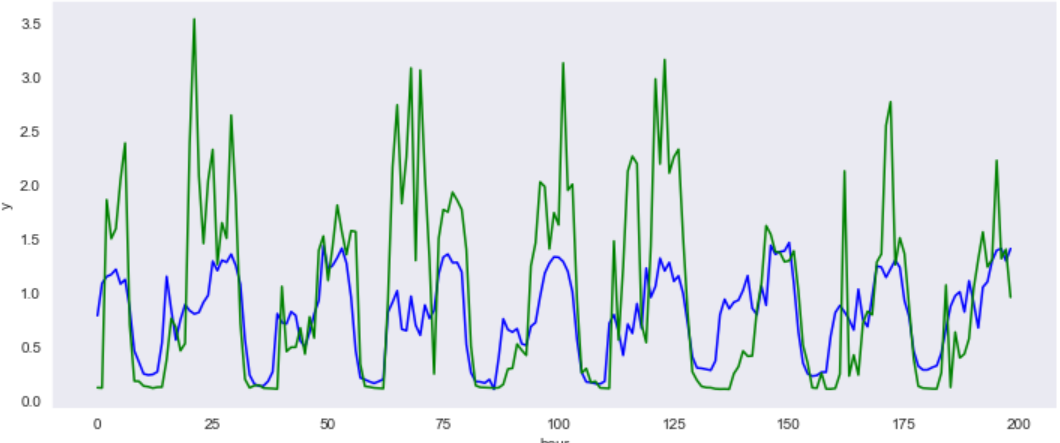
* 1. 对于猜想的验证
     1. 验证策略

由于改进后的方法缺少对于整体电力负载水平的体现，故希望在输入特征中体现每日的电力负载的整体水平。算数平均数可以在一定水平上体现一组连续数据的整体水平，故选择拓展输入特征，将原先只输入前十四日对应时刻的电力负载数据（[1, 14]）拓展成对应数据加上前十四日每日电力负载的平均数值。

* + 1. 验证效果

RMSE = 0.4696

NRMSE = 0.5328



* + 1. 验证分析

从RMSE和NRMSE来看，整体的准确性没有发生太大的变化，从趋势显示图来看，并没有发生太大的效果上的改善。猜测原因如下

1. 平均值并不能够很好的体现每日整体的电力负载消耗水平，要从统计学上寻求更好的表示整体水平的方式
2. 输入特征的处理方式不好，我选择了两种方式进行特征处理，一种是对应时刻与每日平均水平分开放置（即[对应时刻，每日平均值]），一种是二者交叉放置（即[对应时刻，当日平均值，对应时刻，当日平均值，……]），但是都没有表现出很好的峰值预测效果。

所用代码均为Python语言，已经开源至Github仓库

RuidongDavidLin/ML-for-prediction (github.com)

时间2022.10.24-2022.10.31

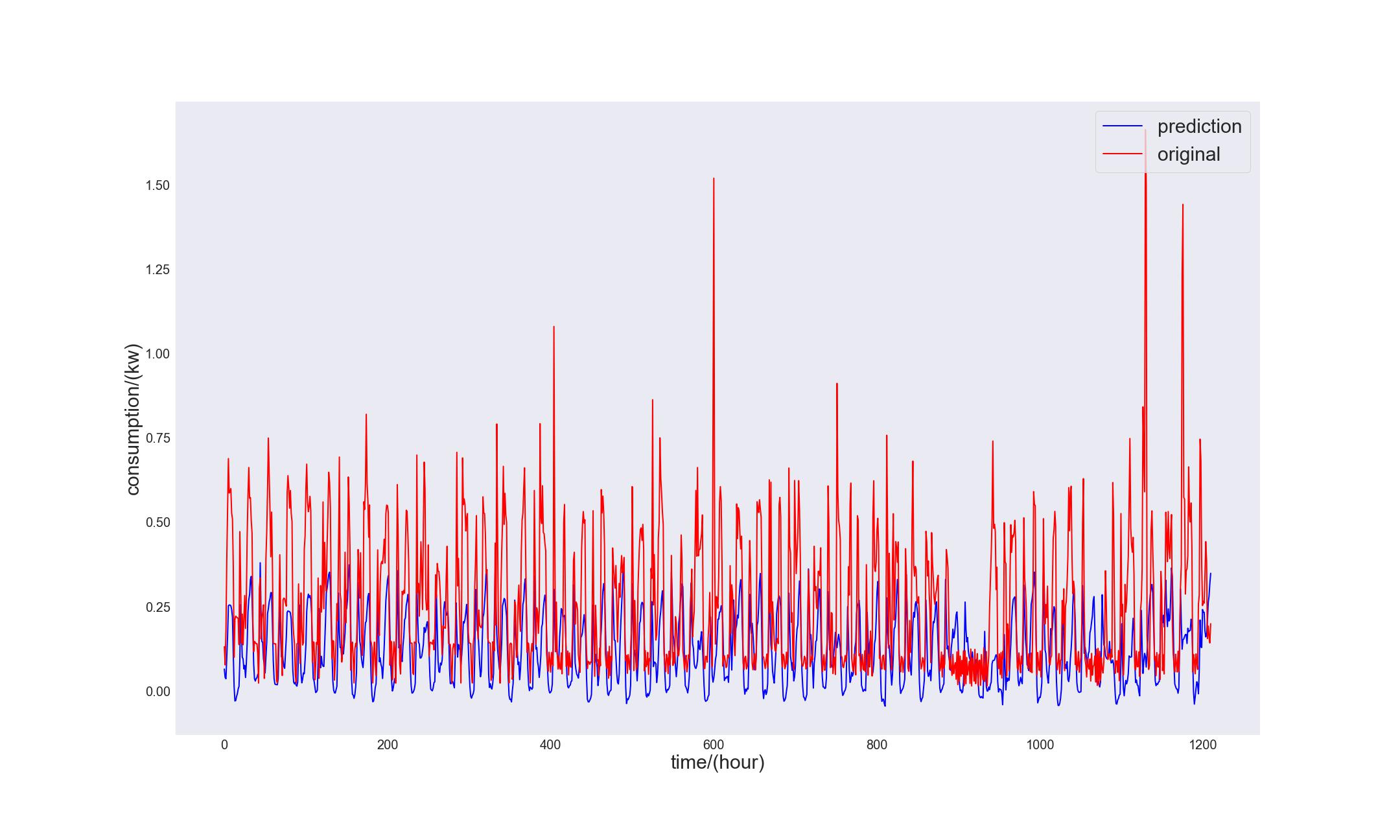
工作内容

1. 根据上周日的组会上，学长和老师提供的建议，对数据特征进行改造。
   1. 对数据进行特征拓展。

考虑到电力负荷具有一定的周期循环性，例如不同天中的同一小时与每周中的同一个星期几都具有一定程度上的行为类似，所以将这两个特征赋予数据。在老师的建议下，我选择利用one-hot encoder（独热编码）对星期几以及每天的不同小时数进行编码（独热编码多用于不具有连续数值特征，而具有分类属性的特征进行编码），因此每个时刻的负载数据相较于原有的数据多了24+7共31维度的信息，然后按照上周的预测方法，组装成一个数据集，每次输入前2天连续时刻的电力负荷数据与每个时刻对应的时间特征（24\*2\*（31+1）共计1536个维度的信息特征）

* 1. 猜想

由于输入的特征维度过大，且独热编码中很多冗余数据（大部分添加的特征维度上都是0），所以将独热编码直接输入模型中进行训练，可能对于模型并不会有特别明显的效果提升

* 1. 实际验证

RMSE=0.6102

NRMSE=0.66304

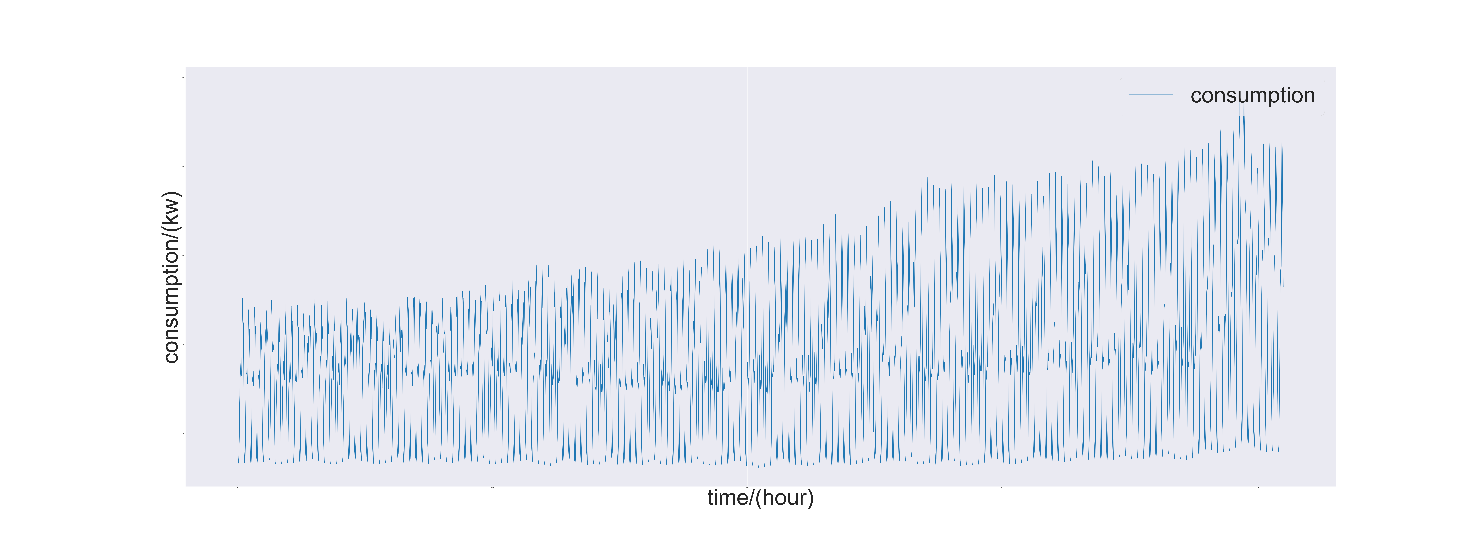
* 1. 分析

实验结果很明显的表示，直接将独热编码作为数据特征输入到神经网络中训练，并不能提升模型性能，甚至相较于原有模型，还有一定准确度上的下降。

经过与老师的讨论与资料的搜索后猜测如下几点原因：

1. 由于是对单个用户做负荷预测，而单个用户的数据波动性较大，同时规律性并不如用户集群那般明显，故预测存在较大的误差。
2. one-hot编码太稀疏了，不要直接输入到神经网络里。正确方法是初始化一个embedding矩阵E，维度是(d, n)，其中n是类别数，d是每个类别的embedding向量。然后用one-hot向量*x*乘***E***取出它对应的那一个embedding向量***Ex***，用这个随机向量输入到神经网络中[1]。
3. 验证猜想
   1. 设计方案

针对老师提出的个体用户行为存在不规律性，这次将数据处理为全体用户的数据分析，将所有用户的数据进行累加，结果如下：

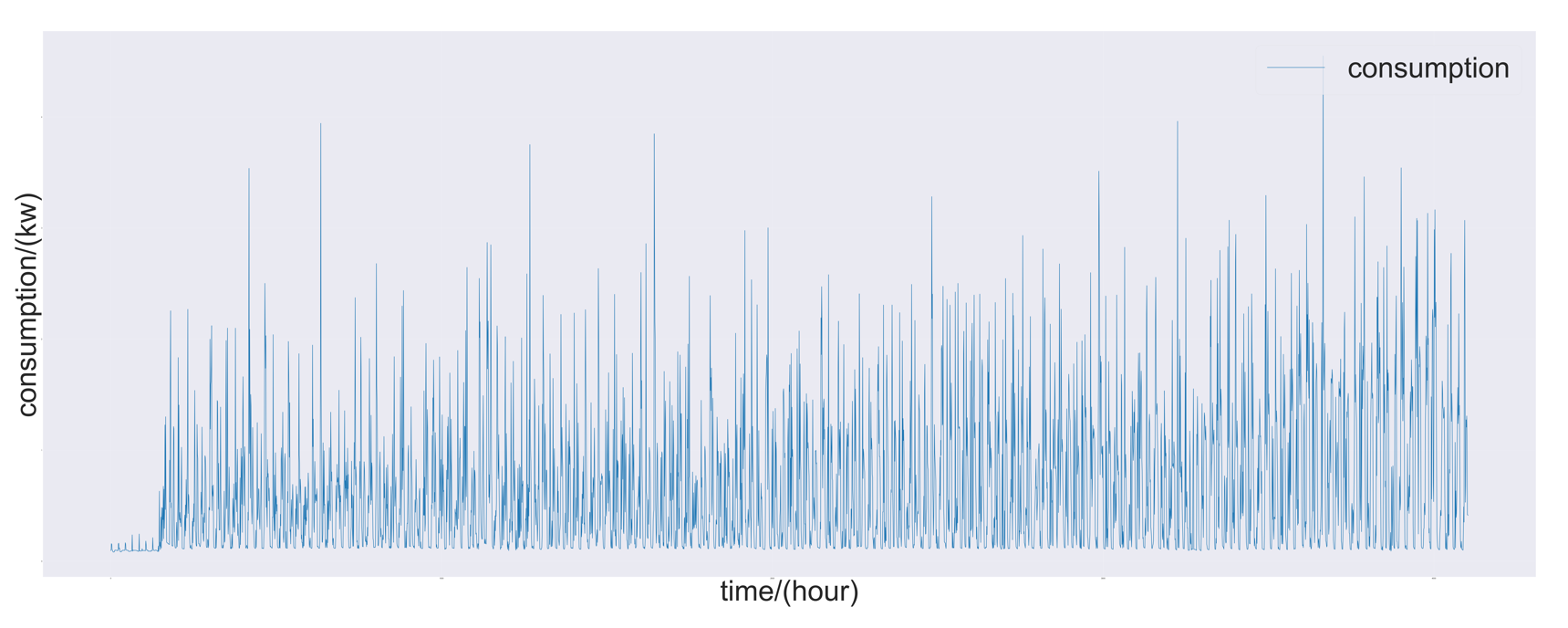


看到集群的电力负荷具有一定的规律性，但是极差过大，不能直接投入到神经网络的训练中；同时，不同用户之间用电模型相差也会比较大，不经分类的直接累加，可能会对用户用电行为模型的识别与预测造成干扰，破坏不同群体用户之间的规律性。

* 1. 改进方法

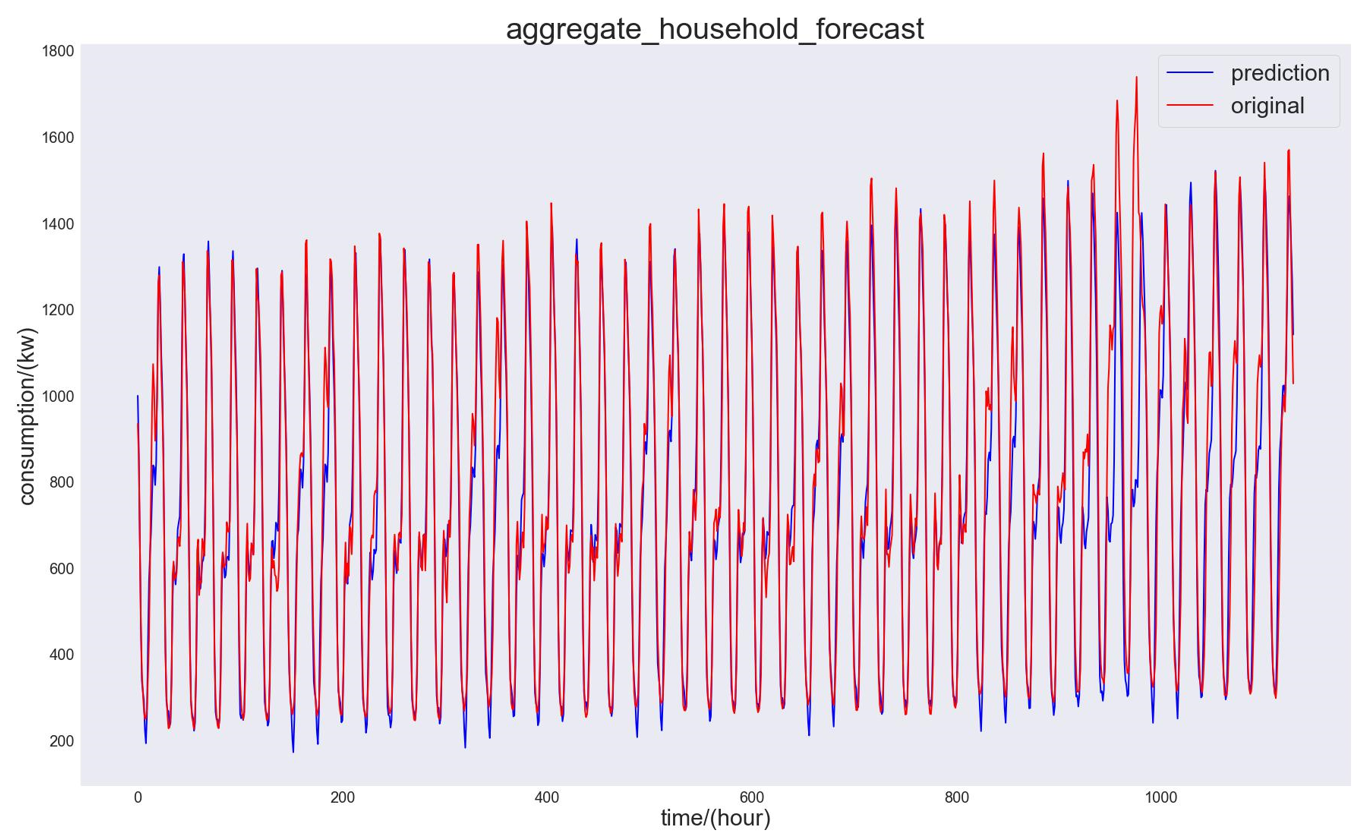
根据老师之前所给的论文，可以用聚类的方式将用户进行分类。

我选择复现老师所给论文里的第一种方式。大致原理为：随机选择K个用户（我选择10个用户）作为聚类的核，同时随机打乱用户顺序。仿照贪心算法选择当前最优解的思想，将每个用户都与选择的聚类核的用户数据进行对比，将用户归到自相关系数提升最大的聚类簇里，从而将3637名用户分类为十种用电行为模式的群体。下面展示为某一群体的累加负载数据：



可以看到数据的极差还是较为明显，无法进行训练，故我选择将数据进行最大值最小值标准化处理。

* 1. 效果验证：



RMSE= 0.1039

NRMSE = 0.1507

可以明显看出，集群聚类后，效果有明显改善，峰值还有趋势预测都明显提升了准确度。

* 1. 实验分析

实验效果并不能很好的在实际应用中展现，因为在数据集处理时，数据标准化是所有数据都标准化处理了，因而对峰值的预测效果较好。虽然在实际应用中，峰值一般也是能在历史数据中有所体现，但是难免会出现极端条件导致出现历史数据中没有体现过的峰值，所以实际应用中，要想有更好的效果，则需要更多维度的数据特征，例如气象预测或者特定时间节点的特征。

[1]作者：godweiyang, 链接：https://www.zhihu.com/question/310017582