金属等能面实验报告（二）

1. 实验问题描述

本实验的目的是根据金属等能面的测量中得到的实空间数据计算出倒空间的数据。根据倒空间到实空间的计算公式：

$$D(\mathbf{r}) = A^3 + 3A|B|^2,\ A = \int d\mathbf{k} f(\mathbf{k}),\ B = \int d\mathbf{k} f(\mathbf{k}) e^{-i\mathbf{kr}} \tag{1}$$

可见，从实空间变换到倒空间，损失了一部分相位信息，要根据倒空间数据计算实空间矩阵，没有直接的解析的逆变换公式，因此，本实验采用机器学习的方法根据对实空间数据进行预测。

2. 实验内容与过程

2.1. 人员分工

所有内容均由一人完成。

2.2. 实验思路

实验采用机器学习的方法，利用训练集提供的数据，对数据进行预处理后进行训练，得到的预处理和预测的模型作用于problem.h5中的输入数据，得到最终预测结果。

此外，实验中也尝试了利用(1)式提供的已知信息，即先计算出训练集target数据对应的系数A，通过机器学习训练出矩阵D与系数A的对应关系，根据此模型对未知数据预测出系数A，然后用|**B**|近似代表**B**矩阵，通过傅立叶逆变换计算出矩阵$f(\mathbf{k})$。然而，由于从实空间到倒空间的信息丢失，此方法最后并不可行。

2.3. 实验过程

- 实验第一步是对输入和输出数据分别进行标准化， 使用的是sklearn库中preproprocessing模块的StandardScaler类，利用训练集的9000组数据生成标准化模型Scaler_feature 和 Scaler_target，并用joblib模块保存模型以便后续使用。此部分代码见文件 train_inputscaler.py 和 train_outputscaler.py；
- 第二步是利用sklearn库中decomposition模块的PCA类分别对输入和输出数据进行特征提取，得到降维后的feature数量为1000维，target数量为2000维。用训练集中的1000个样本对输入和输出数据的PCA模型进行训练并用joblib保存。训练输入数据的PCA模型是整个实验中耗时最长的步骤，大概需要9000秒左右。此部分代码见文件feature_reducetrain.py 和 target_reducetrain.py。然后对训练集的9000组数据均进行标准化和降维，将预处理后的数据保存到h5文件；此部分代码见文件feature_reduce.py 和 target_reduce.py；
- 第三部是进行预测模型训练。所使用的模型是sklearn.kernel_ridge.KernelRidge，kernel='poly'。输入预处理后的feature和target数据进行训练，通过不断地调整模型参数优

化模型，将最后得到的模型用joblib储存。每次训练大概耗时在20秒以内。此部分代码见train.py；

- 最后一步是对问题集里面的数据进行预测。读入之前训练好的KernelRidge模型，标准化模型和PCA模型，先对问题集里面的数据进行预处理，然后用KernelRidge模型进行预测，对预测得到的数据进行PCA和标准化的逆变换，得到最终结果保存于h5文件中。预测用时通常在5秒以内。此部分代码见predict.py。

3. 实验问题讨论

- 实验有待改进的地方，一方面输入和输入参数的维度调整，由于输入数据原规模较大，训练输入数据的PCA模型耗时较长，不便于调整，所以本实验中此参数并没有调到最优。另一方面，预测数据用的模型及其参数还有待探索，可能会有更优的模型或者参数可选。
- 从实空间数据到倒空间数据有解析的表达式可以计算，然而本实验中完全没有利用到这个公式提供的信息，或许可以结合一些公式的推导简化模型。