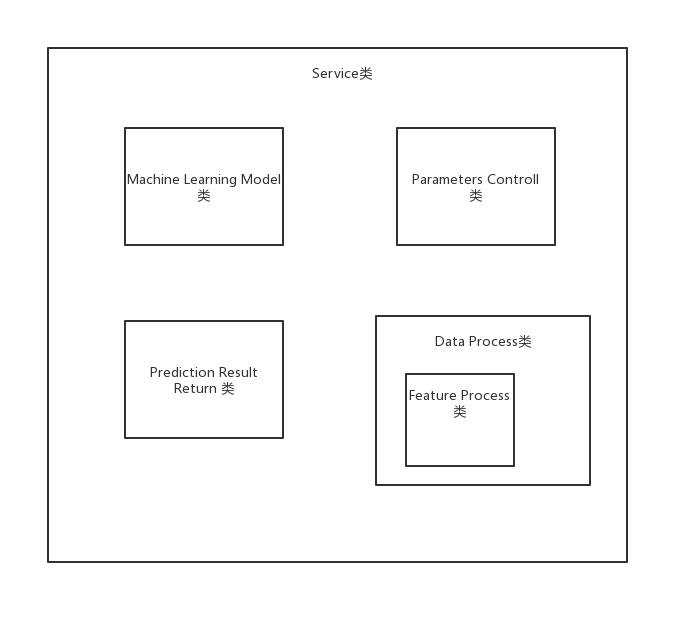
基于机器学习模型的金融预测模型

1. 设计思路



Service类：负责整个系统的业务逻辑，包含Machine Learning类、Parameters Control类、Prediction Result Return 类、Data Process类4大部分。

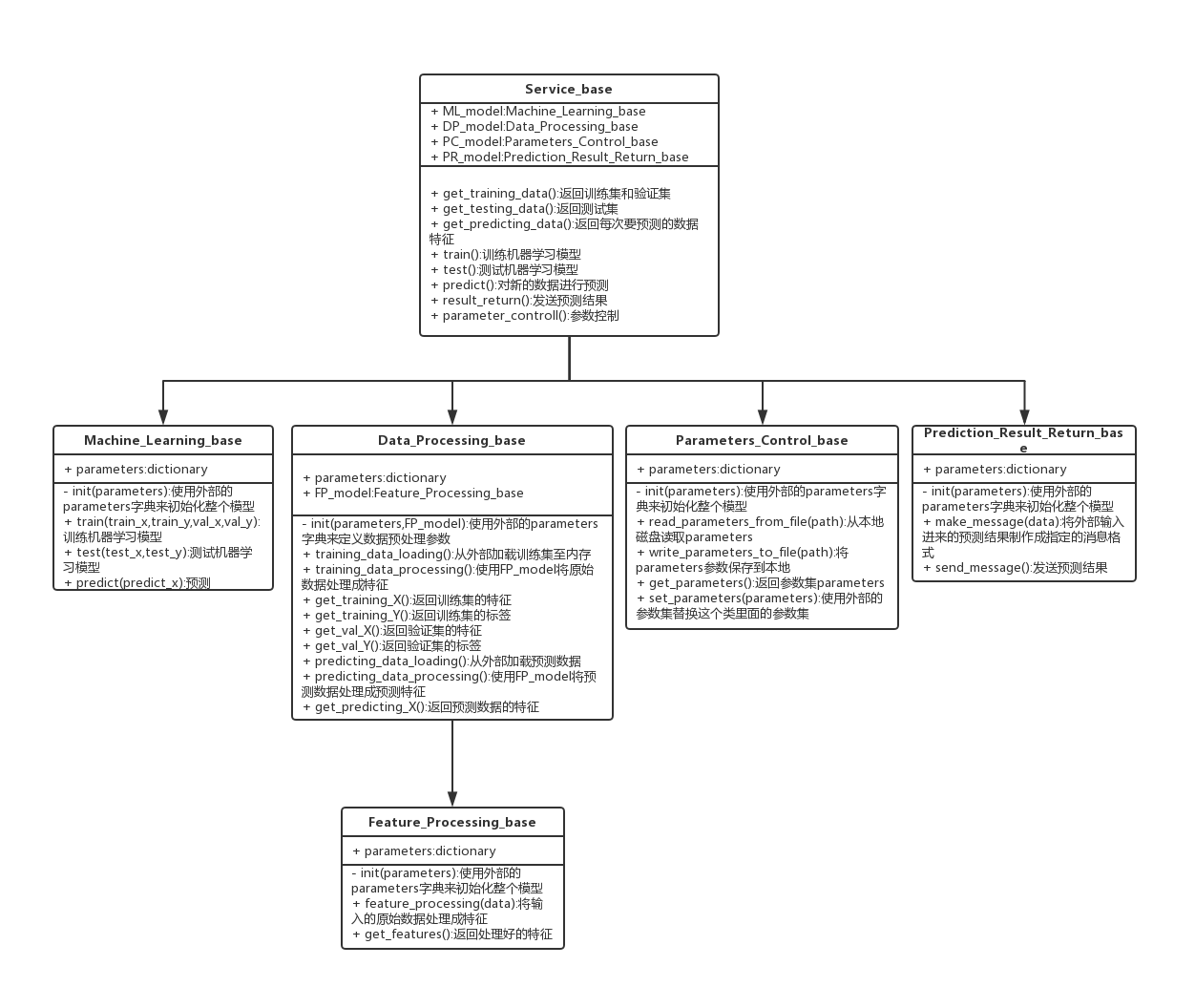
Machine Learning Model 类：包含整个业务所使用的机器学习模型的训练、测试、预测。

Parameters Control类：控制整个模型的各种参数的保存、读取、更新等。

Prediction Result Return 类：将机器学习模型的预测结果发送出去。

Data Processing类：负责读取外部数据，并处理成相应的特征与标签。通过一个特定的Feature Processing对象解耦读取分割数据部分和将raw数据处理成特征部分。

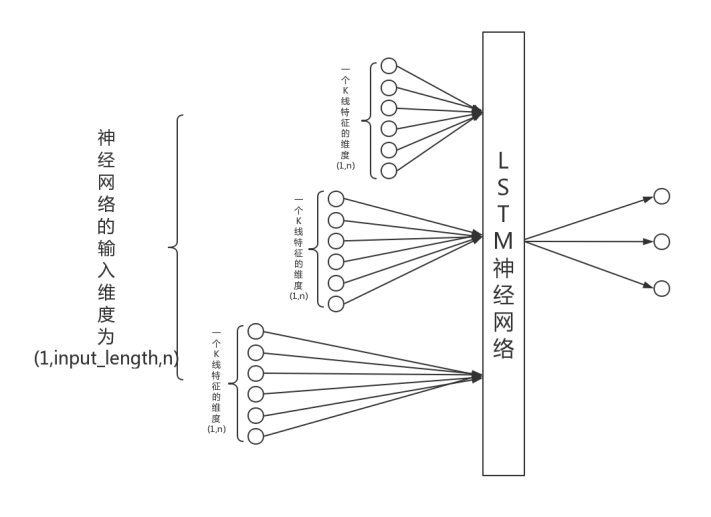
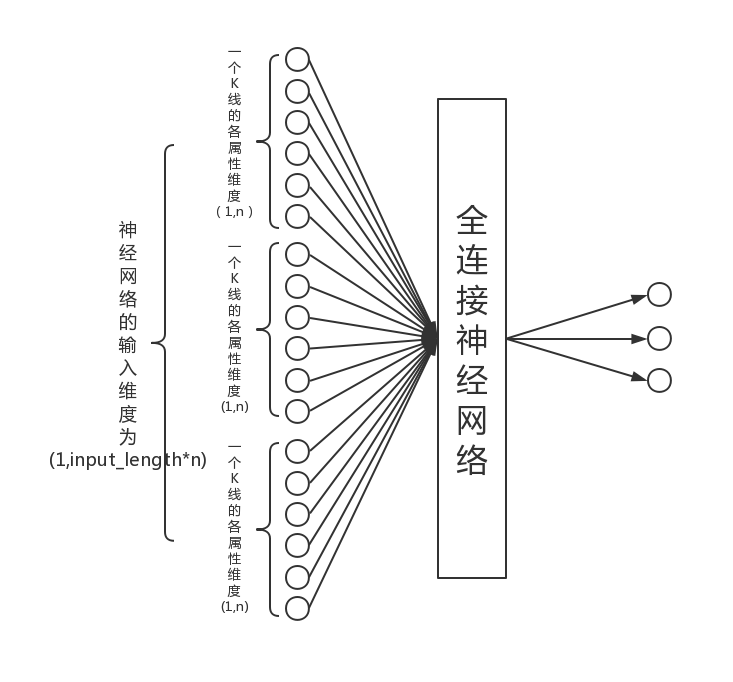
Feature Processing类：负责将数据转化成输入机器学习模型的特征，是Data Processing类的一个成员。

四大模块使用统一的接口进行操作，可多次实现相互替换。而Feature\_Processing\_base作为Data\_Processing\_base的一个组件，在测试多个特征的工程时候可以替换以实现系统的松耦合。

1. Example版的

这个版本是对整个虚基类框架的一个实现，主要是使用神经网络（FC\_Ngram、LSTM-Attention、Self-Attention）进行股票预测。在Machine\_Learning\_Model\_NN.py中，通常读取保存在本地的模型，若没有保存的模型或者模型未训练，则有函数\_\_model\_identify()根据parameters.json里面的参数自动选择模型。

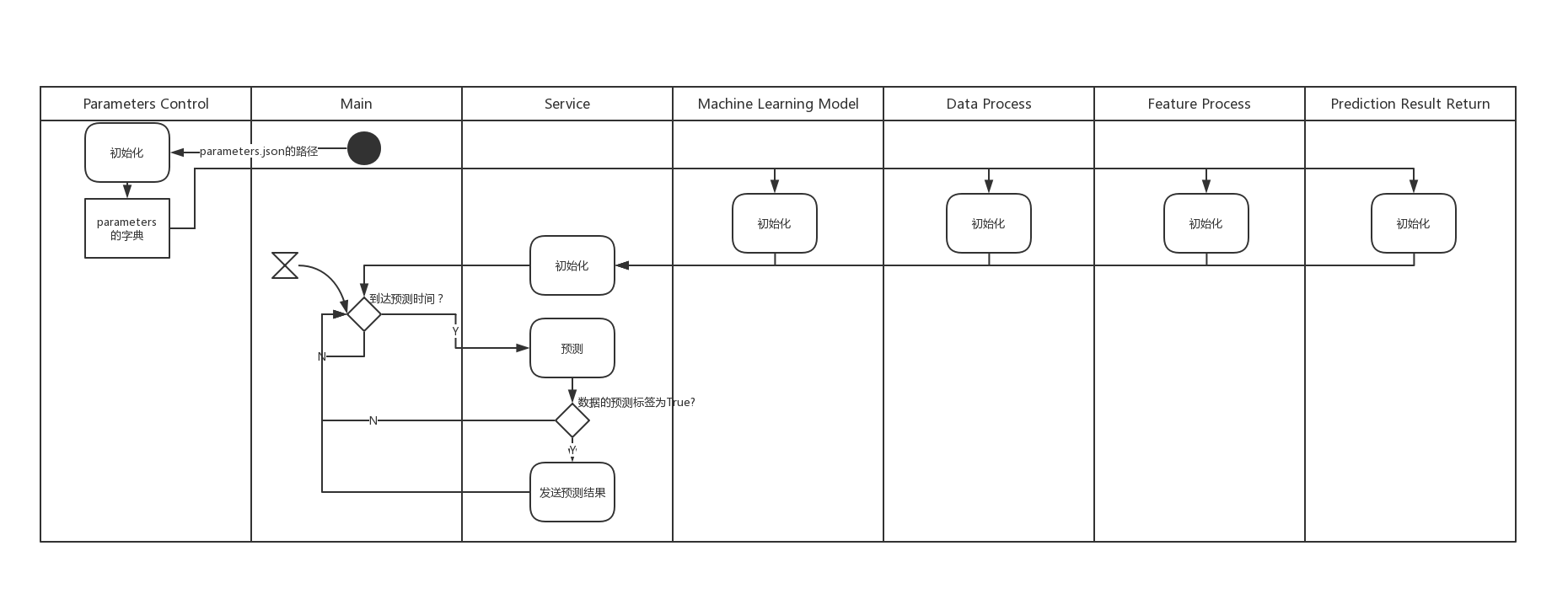
对于可以选择的神经网络模型，共有两种方式进行输入，一种为将一个time\_step的数组全部展开成一列输入的二维数组 ，一种则是常规的三维数组。对于第一种输入仅只有FC\_Ngram模型支持使用。在这两种输入方式中，使用一个参数long\_input控制，当模型为第一种输入方式时long\_input为True,第二种输入方式时为False。



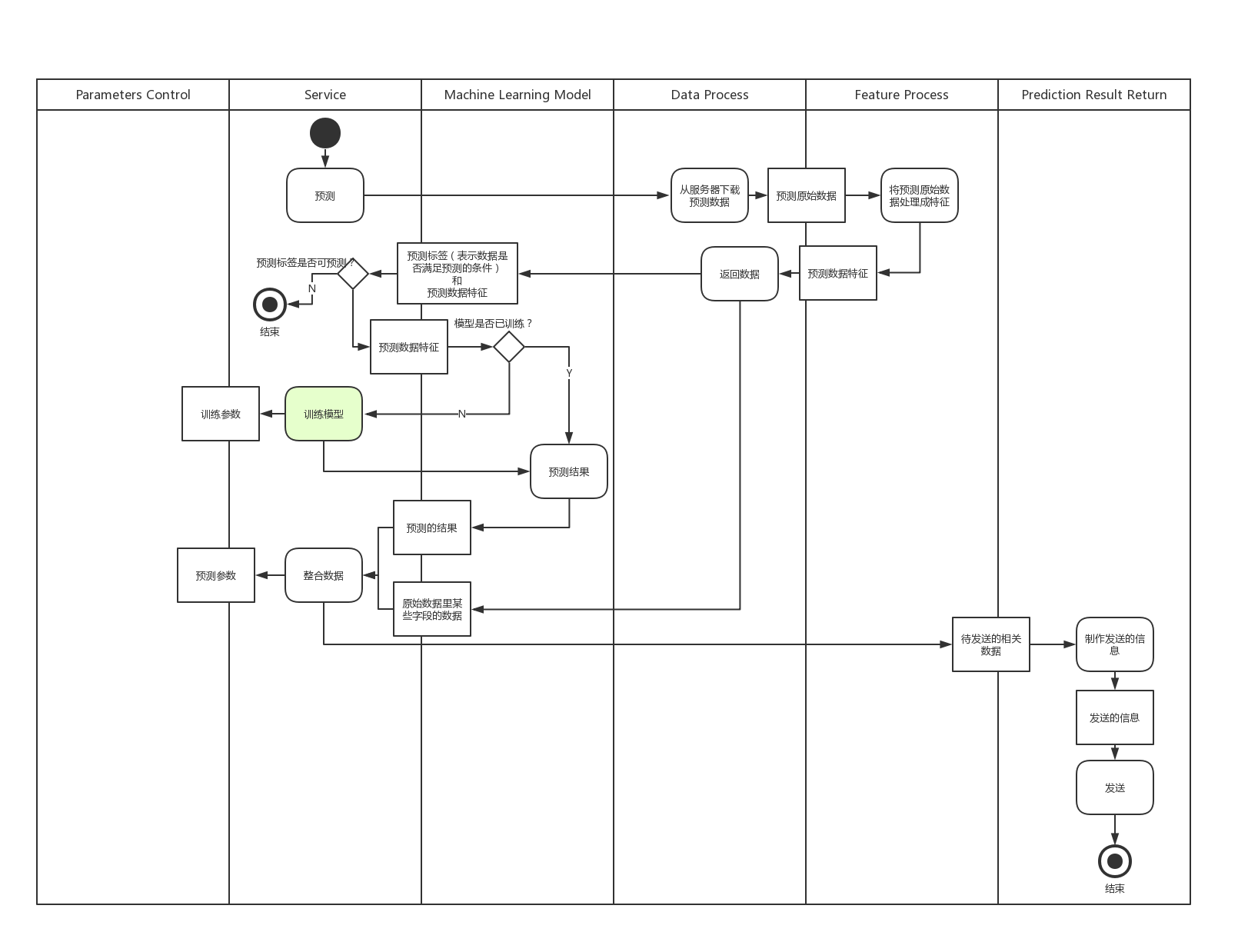
特征工程使用多根K线4个价格之间两两比较和31个标准差两个特征工程完成。在Data\_Processing.py分别读取本地的文件和从服务器上下载数据来训练和预测。对于训练数据，分别获取标签和特征，只将特征输入Feature\_Processing\_MulKCompare.py模块进行特征工程。而在此版本中，由于训练数据和预测数据之间的字段有一些差别，因此在Feature\_Processing\_MulKCompare.py中对于特征工程还区分了训练状态和测试状态。对于以后的版本，如果训练数据和预测数据字段相同，可以使用同一个函数进行特征工程。

在对于参数控制方面，本次的实现使用json文件保存在本地进行读写操作来进行。

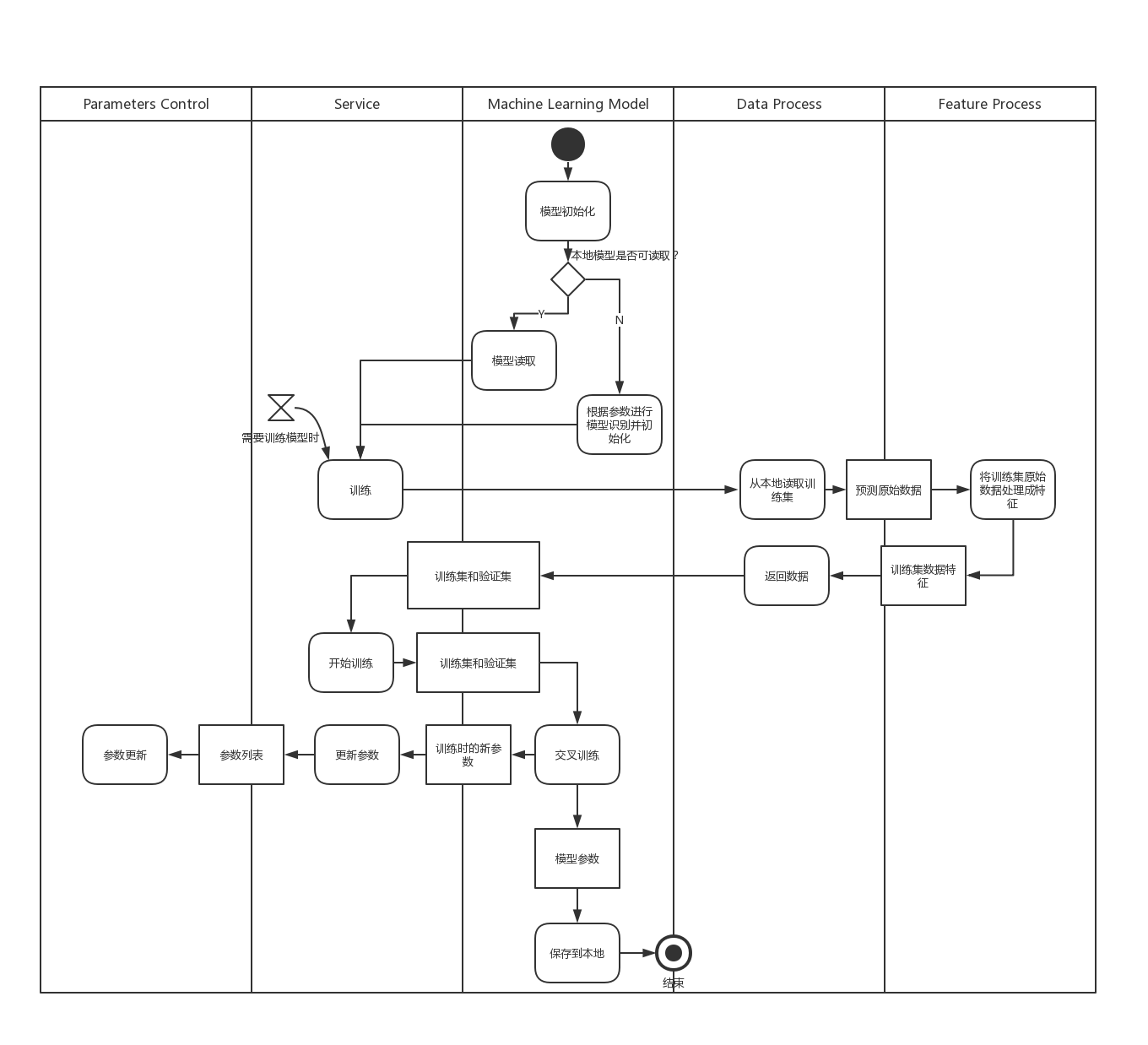
在Service类里面负责整个业务逻辑的控制，其Main函数启动后开始预测，如图所示：



在其中预测这个步骤的逻辑为：



而其中的训练模型步骤的逻辑为：



参数集parameter.json:

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 意义 | 取值范围 | 默认值 |
| model(R) | 决定使用哪个模型 | "Attention\_RNN",“Self\_Attention”,”FC\_Ngram” | 无 |
| long\_input(R) | 对于模型使用长输入还是正常输入 | Bool，如果和model有矛盾会报错 | 无 |
| input\_length(R) | time\_step,决定每一次预测输入多少根K线的特征 | int | 无 |
| hidden\_size(R) | 决定神经网络隐藏层的特征数 | int或int数组，但数组长度需要和层数相等。 | 无 |
| output\_size(R) | 特征的分类类别数 | int | 无 |
| batch\_size(O) | 批训练时每批的大小 | int | 32 |
| rnnmodel(O) | 当模型为Attention\_RNN参数有效，决定采用哪种rnn模型进行模型attention的基础 | ’GRU','LSTM','RNN' | “GRU” |
| activation\_function(O) | 激活函数的选择 | list或者string，'relu','tanh','sigmoid' | relu |
| Feature\_Processing(O) | Data\_Processing的子模块Feature\_Processing的类名 | Feature\_Processing的类名，但需要在Feature\_Processing.py文件里面导入 | Feature\_Processing\_MulKCompare |
| training\_data\_path(R) | 训练数据的路径 | string，可读写路径 | 无 |
| pca\_path(R) | PCA模型保存的路径 | string,可读写路径，以.m为后缀 | 无 |
| pca\_dimension(O) | PCA模型降维的维度 | int | 50 |
| epochs(O) | 神经网络的训练轮数 | int | 120 |
| model\_path(R) | 模型的保存路径 | string,可读写路径，以.pth为后缀 | 无 |
| mul\_K\_compare(O) | Feature\_Processing\_MulKCompare中两根K线之间四个价格两两比较，一根K线与前面一共mul\_K\_compare根K线比较 | int | 13 |
| timeblock(R) | 发送数据时因为允许本系统的服务器和接收方服务器的时区会存在差异，这里将时区差异考虑在内 | Int,小时数 | 无 |
| plateform(R) | 数据的平台(HUOBI) | string | 无 |
| coin(R) | 币种(如BTC) | string | 无 |
| to(R) | 价格(如USDT) | string | 无 |
| timespan(R) | 最低时间尺度数据 | string,”1M”,”1H” | 无 |
| aggregate(R) | 聚合的时间，如15分钟一根K线，则aggregate=15,timespan==‘1M’, | int | 无 |
| con\_regular(R) | 数据获取参数(如"20190402") | string，日期 | 无 |
| con\_simple(R) | 数据获取参数(如"20190402") | string，日期 | 无 |
| send\_lag(R) | 每次预测延后时间 | int | 无 |

带(R)的字段表示必须在初始的参数中给出。带(O)表示可选，若没有则系统选择默认参数。