数据建模的天然气压缩系统

1. 统计学习背景介绍

**统计学习时关于计算机基于数据构建概率模型并运用模型对数据进行预测和分析的一门学科。**

统计学习的对象是数据，它从数据出发，提取数据的特征，抽象出数据的模型，发现数据的知识，又回到对数据的分析与预测中去。

统计学习关于数据的基本假设时同类数据具有一定的统计规律性。统计学习的目的是对数据进行预测和分析，主要过程如下，从给定的、有限的、用于学习的训练数据集合出发，假设数据是独立同分布产生的；并假设要学习的模型属于某个函数的集合，称为假设空间，应用某个评价标准，从假设空间选取一个最优的模型，使它对已知训练数据及未知测试数据在给定的评价准则下有最优的预测；最优模型的选取由算法实现。

1. 算法学习
2. SVM

Svm因为强大的非线性拟合、特征映射、不过拟合等有点广泛应用于工业界等数据获取难度高、实时性较强等场景。

[支持向量机通俗导论（理解SVM的三层境界）](https://blog.csdn.net/v_JULY_v/article/details/7624837)

[统计学习方法，李航著](https://book.douban.com/subject/10590856/)

[机器学习，周志华著](https://book.douban.com/subject/26708119/)

1. 随机森林

针对多个弱学习器与强学习器的关系（集成学习），集成学习是使用一系列学习器进行学习，并使用某种规则把各个学习结果进行整合从而获得比单个学习器更好的学习效果的一种机器学习方法。一般情况下，集成学习中的多个学习器都是同质的"弱学习器"。

随机森林对噪声和异常值有较好的容忍性， 对高维数据分类问题具有良好的可扩展性和并行性。

[Random Forest's homepage (by Leo Breiman and Adele Cutler)](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#inter)

[Ensemble learning](http://www.scholarpedia.org/article/Ensemble_learning)

[Introduction to Random forest - Simplified](https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/)

1. 数据建模流程图

训练数据获取、清洗、过滤。

数据预处理、构建特征空间。

利用算法挑选最优模型（SVM、RandomForest）。

初始化算法参数。

预测数据获取、清洗、过滤。

保存算法参数

加载训练好的算法及参数。

得到预测结果。

图1、数据建模流程图

其中全部实现代码在Windows10，Vs2015与Matlab2018a调试验证通过，考虑到并未使用与其他版本不兼容代码，Windows7及以后、Vs2013及以后、Matlab2014及以后应均可运行。

考虑到代码代码保护、上下位机通信、代码维护等情况，以上全部流程均以c++编写，Matlab仅作为dll（动态链接库）的调用环境，具体实现过程见下文。

1. 具体实现过程
2. 代码依赖环境安装

考虑到算法实现过程包含大量数学与矩阵运算，特安装opencv3.4.3库及stl标准库用以加速算法实现及运算速度。（其中stl库编译器自带，故只考虑opencv3.4.3安装过程）。

（默认已经安装Visual Studio及Matlab）。

Opencv3.4.3下载链接：<https://opencv.org/releases.html>

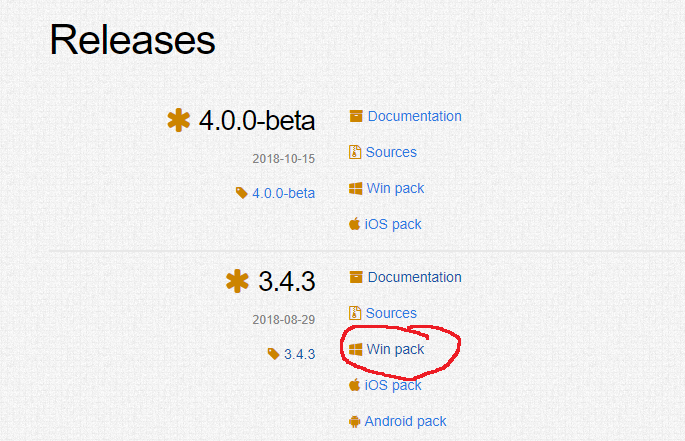


图2、opencv3.4.3下载链接

下载完毕，双击文件进行解压。

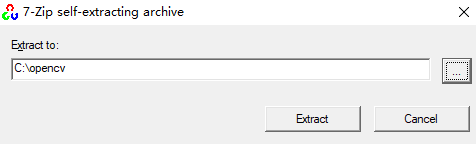


图3、opencv解压

在右击此电脑->属性->高级系统设置->环境变量->系统变量->Path->添加（C:\opencv\opencv\build\x64\vc14\bin）。

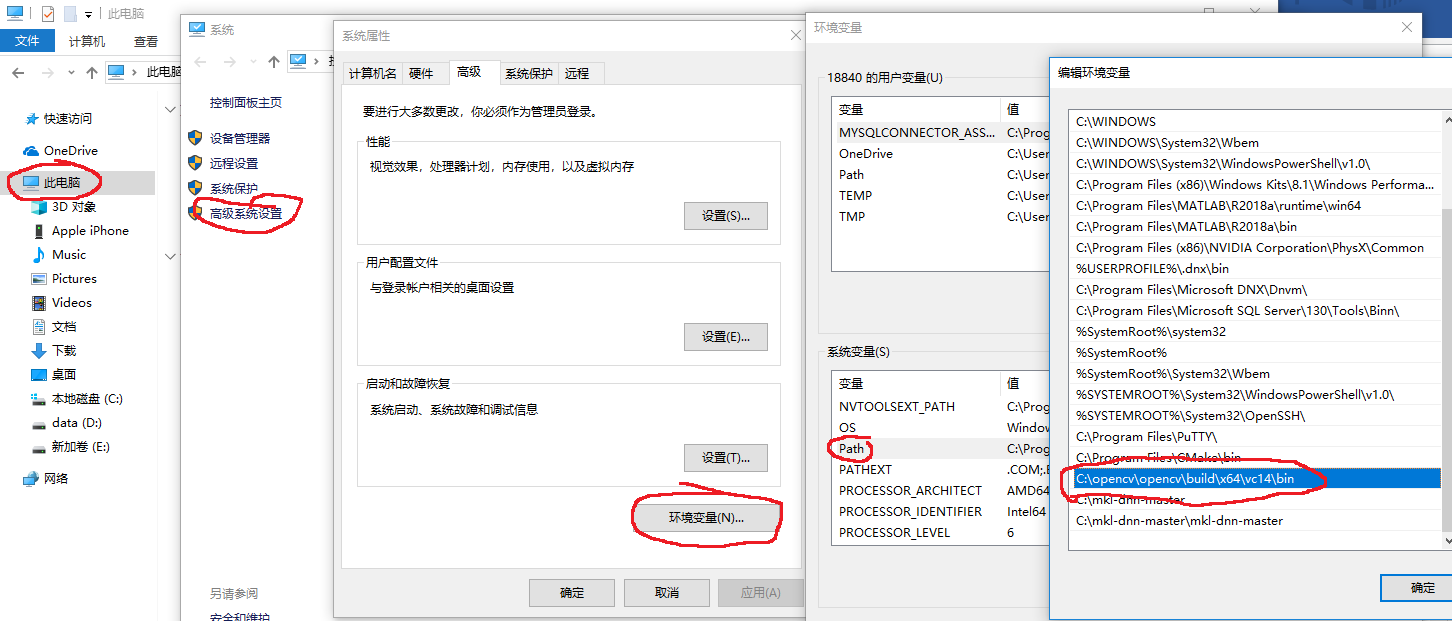


图4、添加环境变量

在项目中添加opencv路径：打开Vs，新建一个项目（或从已有文件中建立项目），File->New->Project(Project from existing code)->Visual C++->Win32 Console Application ->(选择好项目路径及名称后点击ok)->Next->Finish。

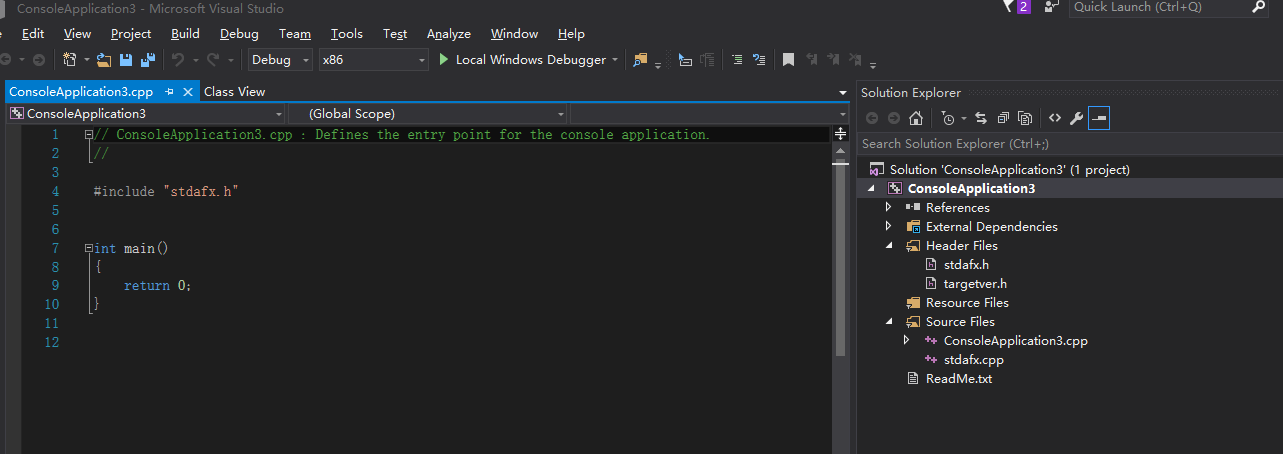


图5、建立Vs项目

右击项目名称（此处为ConsoleApplication3），选择Properties->C/C++->General->Additional Include Diretories->

(C:\opencv\opencv\build\include

C:\opencv\opencv\build\include\opencv

C:\opencv\opencv\build\include\opencv2)

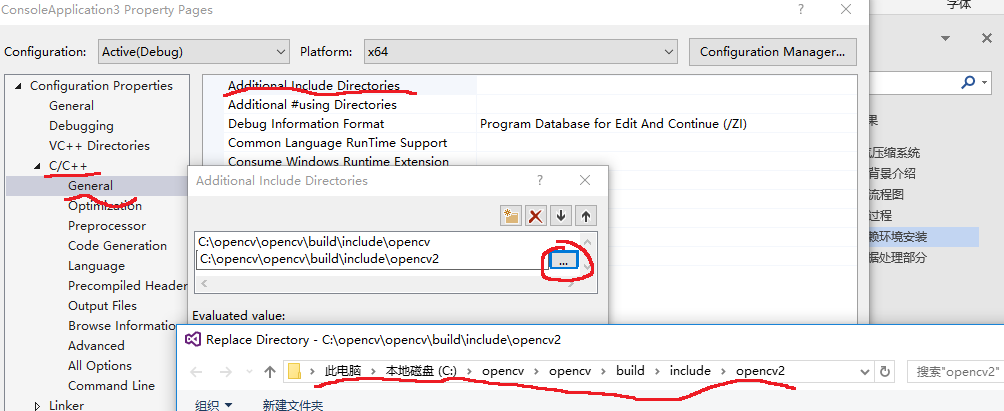
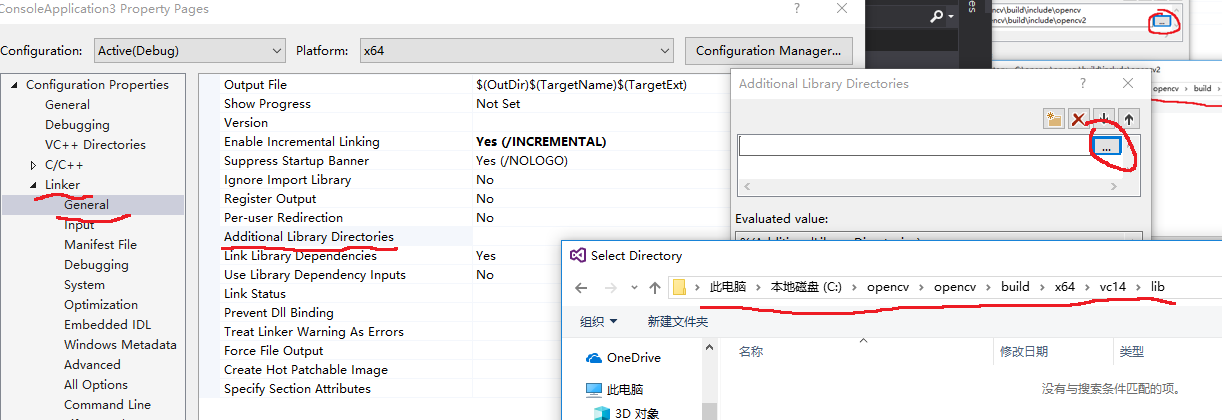


图6、添加opencv库地址

添加opencv静态链接库地址，Linker->General->Additional Library Directories->(C:\opencv\opencv\build\x64\vc14\lib)

Vs2015选择vc14，Vs2017选择vc15。

Linker->Input->Additional Dependencies->opencv\_world343d.lib



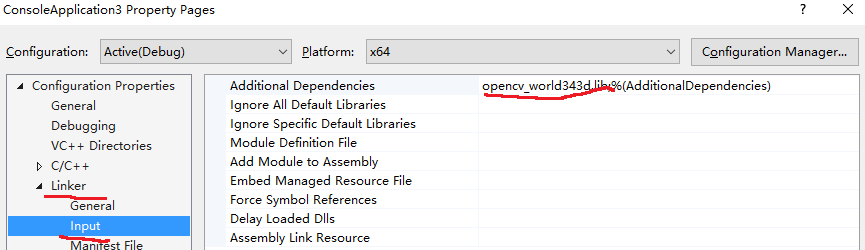


图7、插入静态链接库

Opencv测试，将编译器环境改为64位。运行如下代码，图片路径自定义。

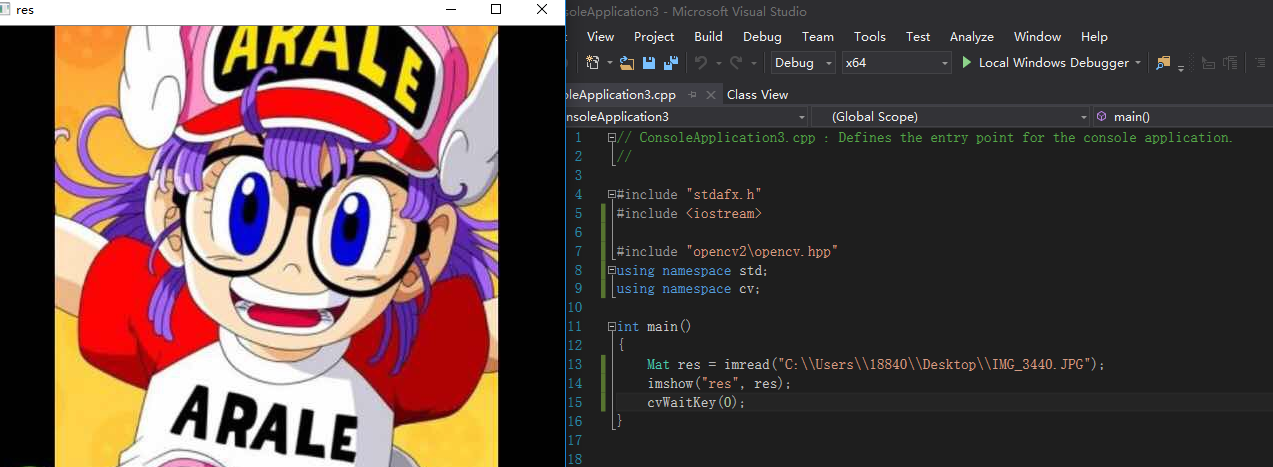


图8、opencv测试

1. 文件夹结构

C++Project

|--interface.h (各个模块具体代码)

|--getData.h

|--preProcess.h

|--model.h

|--svmModel.h

|--

…….

|--data （请确保为csv文件）

|--data1.csv

|--data2.csv

|--log （用以保存日志，具体输出见下文）

|--model （保存算法系数）

|--SVM

|--RandomForest

Matlab Project

|--data （包含数据文件）

|--log （包含日志文件）

|--model （包含模型文件）

|--dllInterface.cpp （各个代码）

|--

….

注：该软件会自动调用project下的data文件夹下全部csv文件，训练好的模型放在model文件夹下，日志文件包含debug信息，在log文件夹下。

1. C++代码整体框架

因为实现dll库的封装，故留下train、loadModel、predict三个接口，在interface.h文件下。

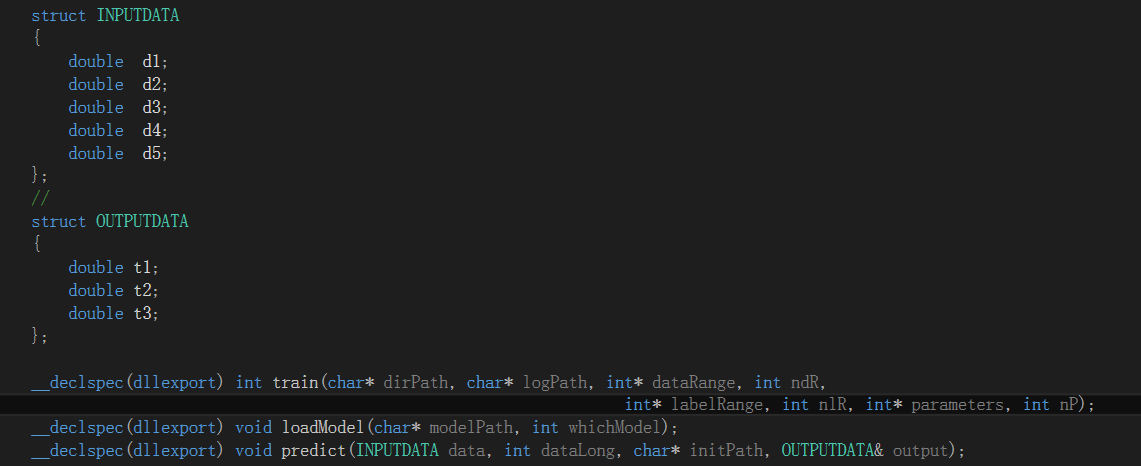


图9、interface.h文件

其中各个参数、接口作用如下：

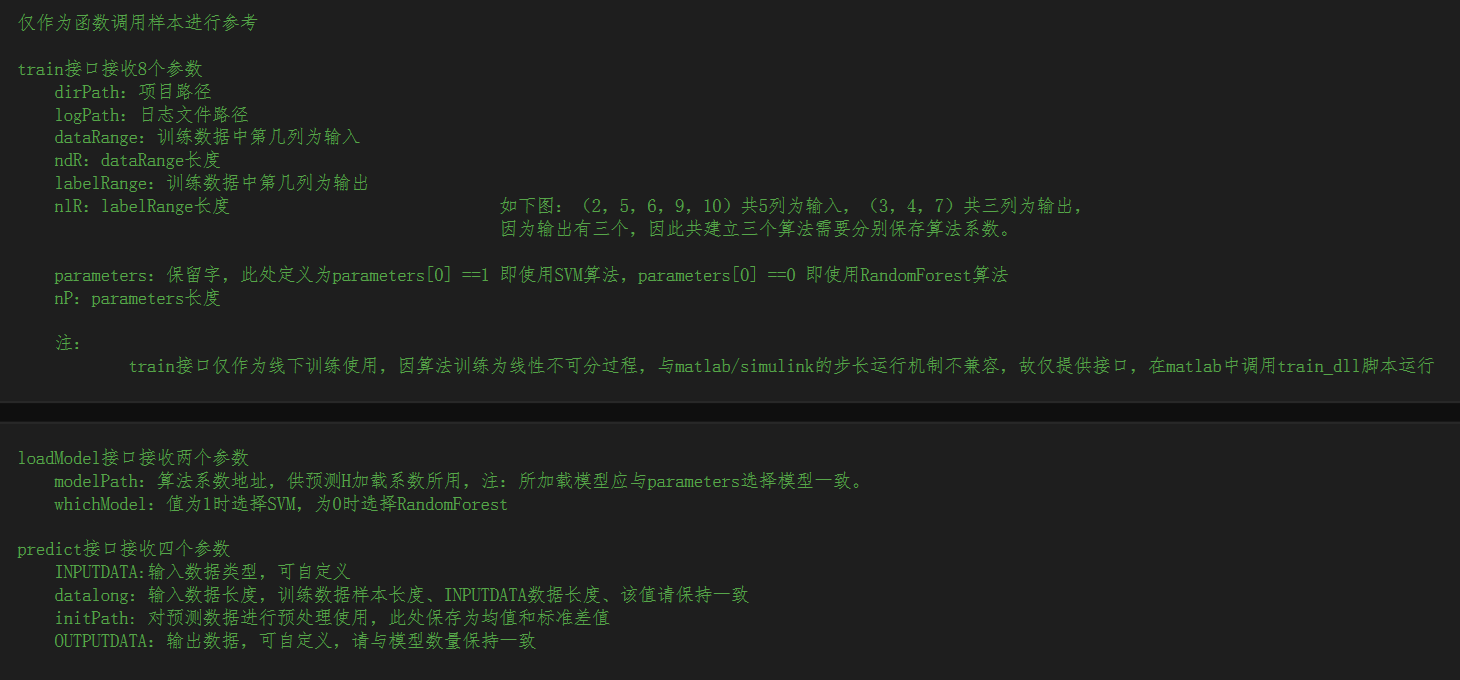


图10、接口输入输出参数说明

还包含数据处理模块（getData.h、preProcess.h）、算法实现模块（model.h、svmModel.h、randomForest.h）、额外可添加模块（util.h）。下文进行详细说明。

3.1、数据处理部分

主要代码为getData.h、preProcess.h.

getData.h部分自动识别dirPath下的全部csv文件，并读取dataRange和labelRange的数据，主要流程为，加载全部csv文件->读取数据（此时为字符串格式）->float格式->过滤清洗->预处理。

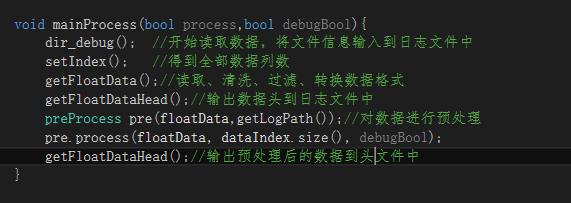


图11、读取数据全部过程

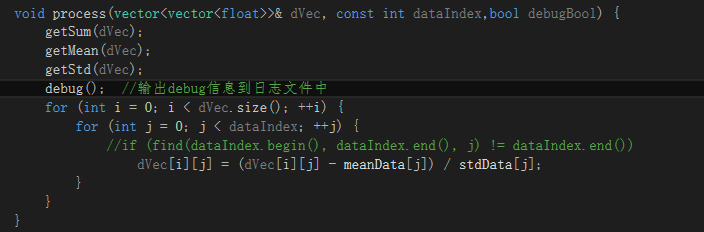


图12、预处理过程

3.2、算法实现部分

主要代码为model.h、svmModel.h、randomForest.h。

为实现向后兼容性，model.h为纯虚类，各个函数见文知意。

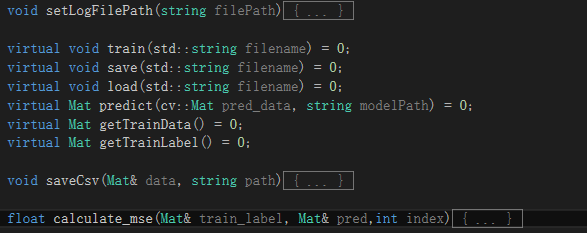


图13、model.h文件

svmModel.h为支持向量机算法具体实现过程，randomForest.h为随机森林具体实现过程。

3.3、额外可添加模块

Util类主要是为了对数据进行区分和输出日志文件用。

之后再model文件夹、log文件夹下生成如下所示文件

Log

|-- 11M\_12D\_17\_H （月\_日\_时）

|-- 17H\_12M\_44S.txt （时\_分\_秒）

Model

|--SVM

|-- 11M\_12D\_17\_H （月\_日\_时）

|-- 17H\_12M\_44S.xml （时\_分\_秒）

|--RandomForest

|-- 11M\_12D\_17\_H （月\_日\_时）

|-- 17H\_12M\_44S.xmlt （时\_分\_秒）

3.4、C++部分使用详情

C++代码可独自运行也可生成dll库供matlab调用，现详细介绍两个流程。

独自运行exe文件：右击项目文件->Properties->Configuration Properties->General->Configuration Type->exe

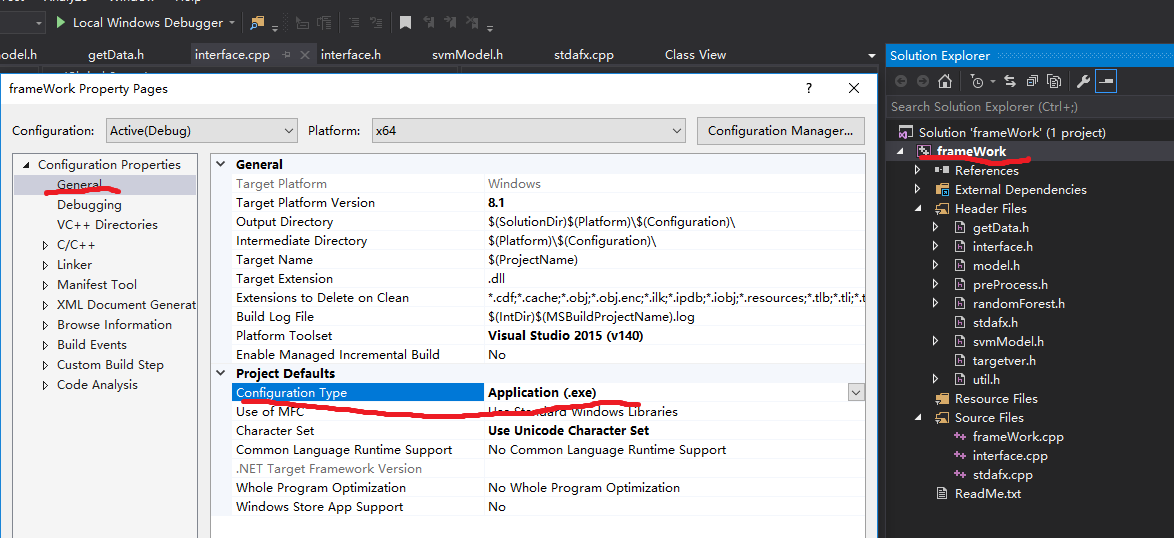


图14、运行exe配置

注释掉interface.cpp文件，只保留#include “stdafx.h”这一配置命令

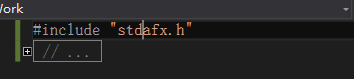


图15、interface.cpp注释

保留frameWork.cpp文件，按照需求在int main（）函数里修改，之后点击ctrl+f5，即可在log文件下发现相应的debug信息。

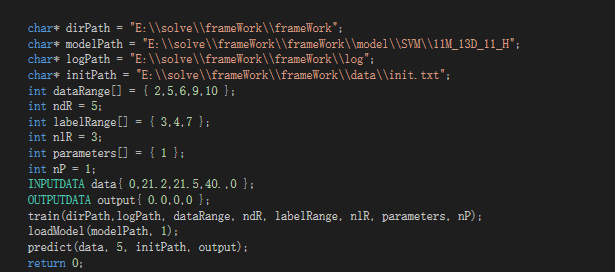


图16、frameWork.cpp文件

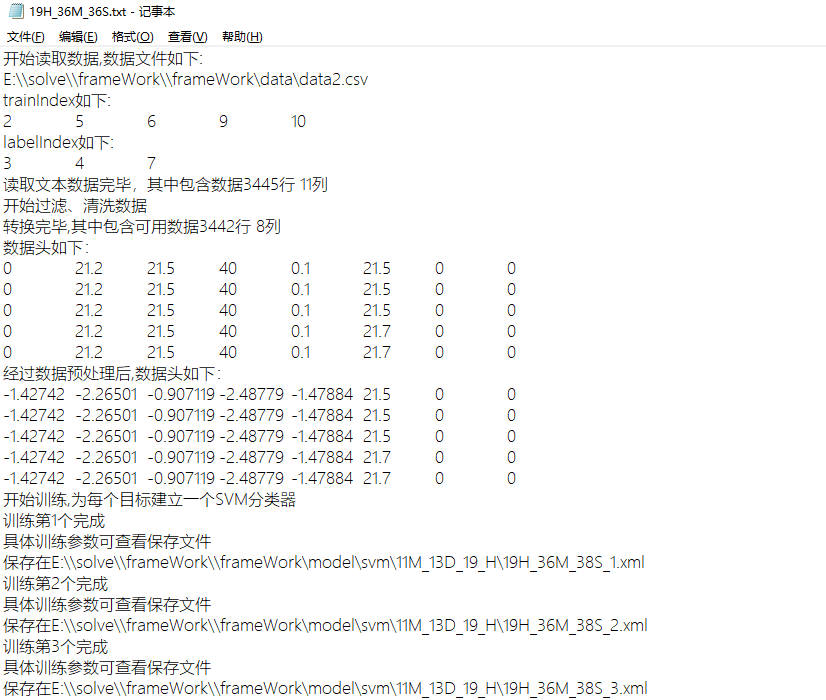


图17、log日志文件debug信息

生成dll文件，因考虑到之后可能修改输入输出参数个数或其他更改dll文件的情况。修改dll需手动更改interface.h文件的接口和interface.cpp文件内容。这里需将frameWork.cpp内容注释仅保留#include “stdafx.h”，反注释interface.cpp文件。同时修改属性为dll。

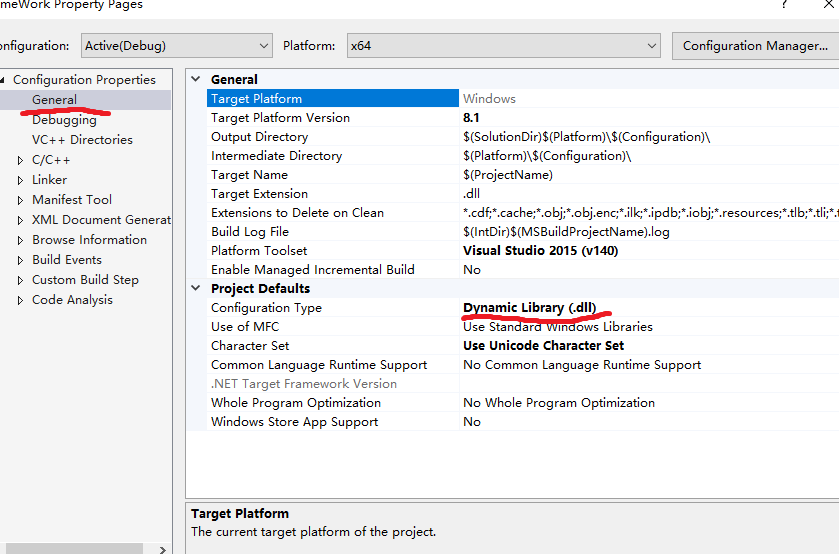


图18、dll生成配置信息

在修改且配置结束后，点击ctr+alt+f7，即可在项目路径下->x64->Debug发现frameWork.dll文件，将其移到Matlab Project文件夹下进行替换，并修改相应的matlab配置信息如 NUMIN NUMOUT…，再对Matlab Project下的train\_dll.cpp 、 dllInterface.cpp重新编译即可（mex train\_dll.cpp）。

1. Matlab代码整体框架

4.1、数据训练部分

主要通过train\_dll.cpp文件作为调用dll的接口，之后在matlab命令窗口 mex train\_dll.cpp回车即可生成train\_dll.mexw64文件。



图18、train\_dll.cpp文件

具体使用方式参见exampleTrain.m文件

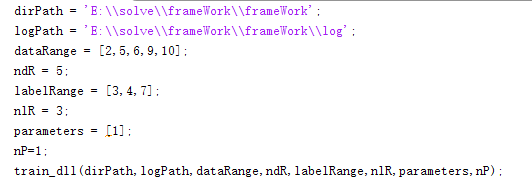


图19、exampleTrain.m文件

4.2、数据加载和预测部分

此部分使用simulink s function实现，具体代码在dllInterface.cpp，根据需要修改输入个数和输出个数，其中S-function-parameters该修改步长时间。

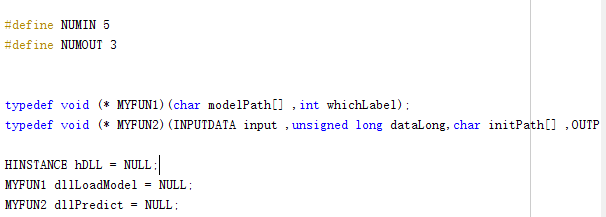


图18、dllInterface.cpp文件

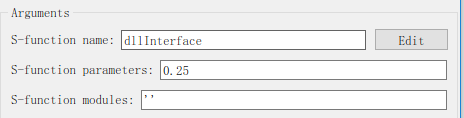


图19、S-function属性

具体使用方式见examplePredict.m文件。simulink模型文件见loadModelAndPred.mdl文件。

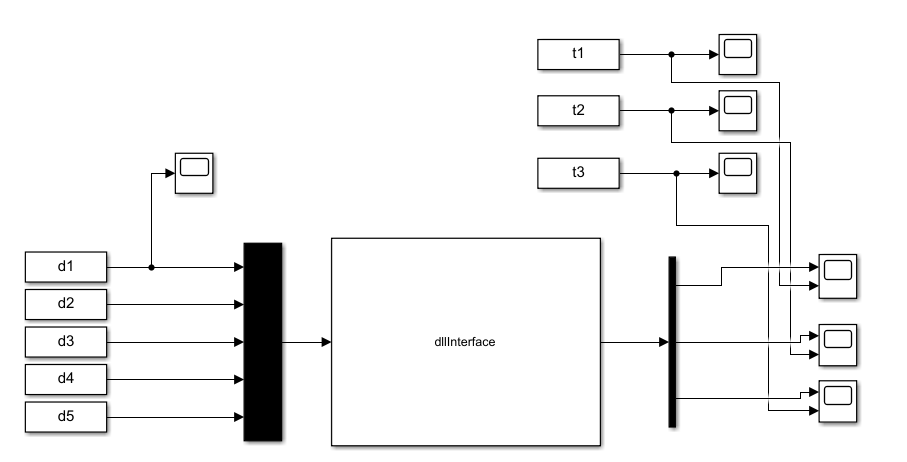


图20、loadModelAndPred.mdl文件

1. 样例结果展示

所给数据为艾默生10月测试数据.xlsx，手动拆分为data1.csv、data2.csv、data3、csv，经沟通得知进气压力、油分温度、吸气温度、频率、天然气量是输入，排气温度、出口压力、最终排气压力是输出。