

# 一、概述

1、以鸢尾花分类为例，采用TensorFlow2.0训练模型并保存模型pb格式，然后将pb文件转换为onnx文件，再用倍福ModelManager工具将onnx文件转换为xml文件，再用tc3调用xml，输入鸢尾花花萼长度、花萼宽度、花瓣长度、花瓣宽度输出预测的分类。

2、测试环境：win10 64位，TensorFlow2.0，tc3 v3.1.4024.10

# 二、环境输入搭建

## 1、TwinCat环境

安装插件TF38xx-Machine-Learning。本教材TwinCat3版本，v3.1.4024.10

## 2、TensorFlow2.0环境

参考：<https://www.bilibili.com/video/BV1B7411L7Qt?from=search&seid=7944276473719331262>

# 三、iris数据保存

后期需要验证，所以将鸢尾花分类数据保存到一个csv文件，iris.csv

```
from sklearn import datasets
from pandas import DataFrame
import pandas as pd

x_data = datasets.load_iris().data # .data返回iris数据集所有输入特征
y_data = datasets.load_iris().target # .target返回iris数据集所有标签
print("x_data from datasets: \n", x_data)
print("y_data from datasets: \n", y_data)

x_data = DataFrame(x_data, columns=['花萼长度', '花萼宽度', '花瓣长度', '花瓣宽度']) # 为表格增加行索引（左侧）和列标签（上方）
pd.set_option('display.unicode.east_asian_width', True) # 设置列名对齐
print("x_data add index: \n", x_data)

x_data['类别'] = y_data # 新加一列，列标签为‘类别’，数据为y_data
print("x_data add a column: \n", x_data)

irisDataFrame=pd.DataFrame(data=x_data)

csvPath='iris.csv'      #str(dateTime)
irisDataFrame.to_csv(csvPath,header=True,encoding="gbk")
```

# 四、模型训练及保存

## 1、python代码

```
import tensorflow as tf
from sklearn import datasets
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd

#iris数据下载
x_train=datasets.load_iris().data
y_train=datasets.load_iris().target

print('x_train: ',x_train)
print('y_train: ',y_train)

#数据打乱
np.random.seed(116)
np.random.shuffle(x_train)
np.random.seed(116)
np.random.shuffle(y_train)
tf.random.set_seed(116)

#神经网络搭建
model=tf.keras.models.Sequential()

[tf.keras.layers.Dense(3,activation=tf.nn.softmax,kernel_regularizer=tf.keras.regularizers.l2())
]

model.compile(
    optimizer=tf.keras.optimizers.SGD(lr=0.1),
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
    metrics=tf.keras.metrics.sparse_categorical_accuracy
)

model.fit(x_train, y_train, batch_size=32, epochs=500, validation_split=0.2, validation_freq=20)

model.compile()
# 模型保存
model.save('saved_model/my_model')
```

## 2、保存的模型

saved_model > my_model		
Name	Date modified	Type
assets	2020/8/24 17:21	File folder
variables	2021/4/6 14:23	File folder
<b>saved_model.pb</b>	2020/9/28 10:26	PB File

## 五、模型转换

### 1、模型pb文件转onnx文件

参考: [https://blog.csdn.net/rookie\\_wei/article/details/99448905](https://blog.csdn.net/rookie_wei/article/details/99448905)

### 4、将tensorflow模型转换成ONNX模型

先安装tf2onnx模块，执行下面命令，

```
pip install -U tf2onnx
```

接着，就要将上面的模型转成ONNX的模型，用一个命令行就可以搞定了，命令如下，

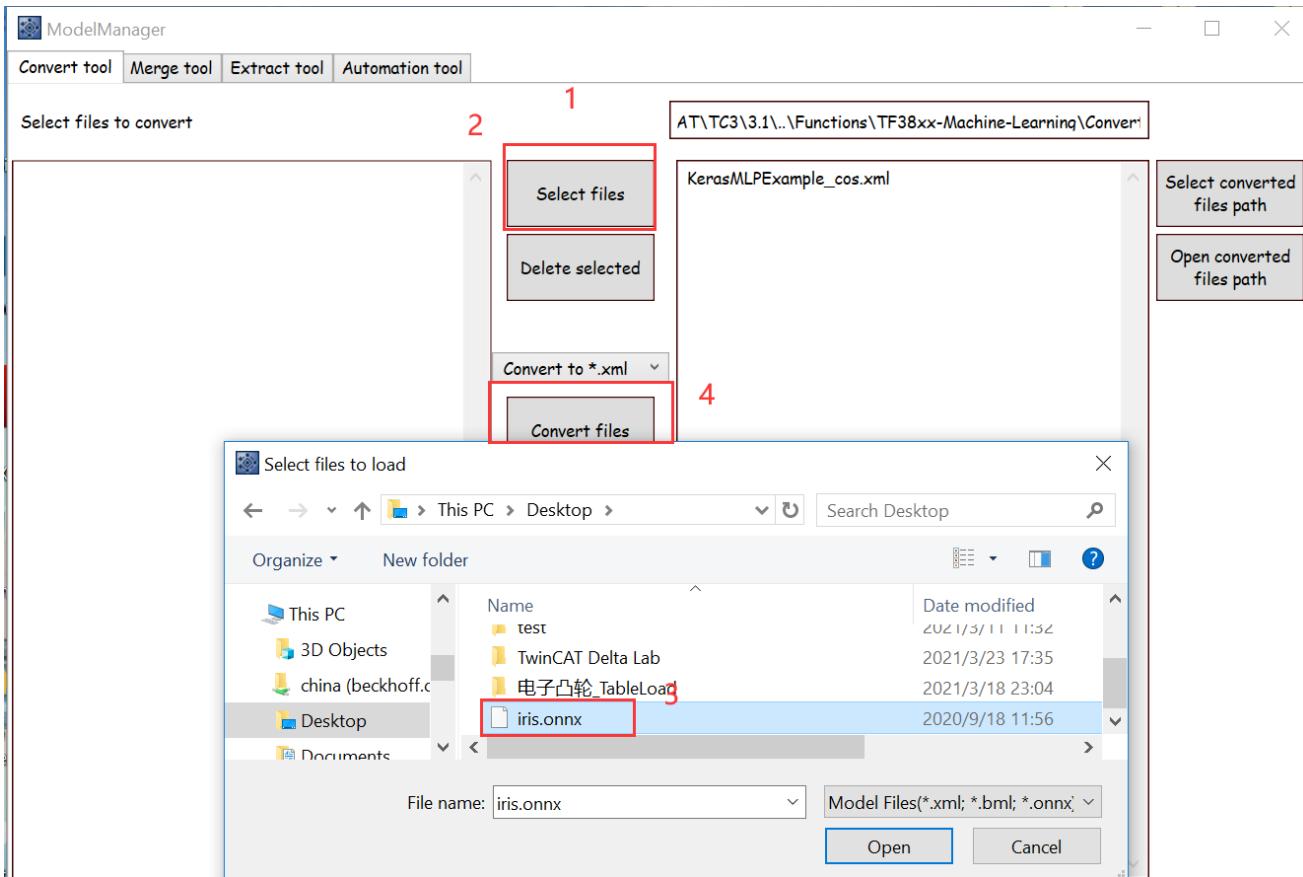
```
python -m tf2onnx.convert --saved-model ./saved_model --output ./mnist_onnx/mnist.onnx --opset 7
```

onnx文件

saved_model > iris_onnx		
Name	Date modified	Type
<b>iris.onnx</b>	2020/9/18 11:56	ONNX File

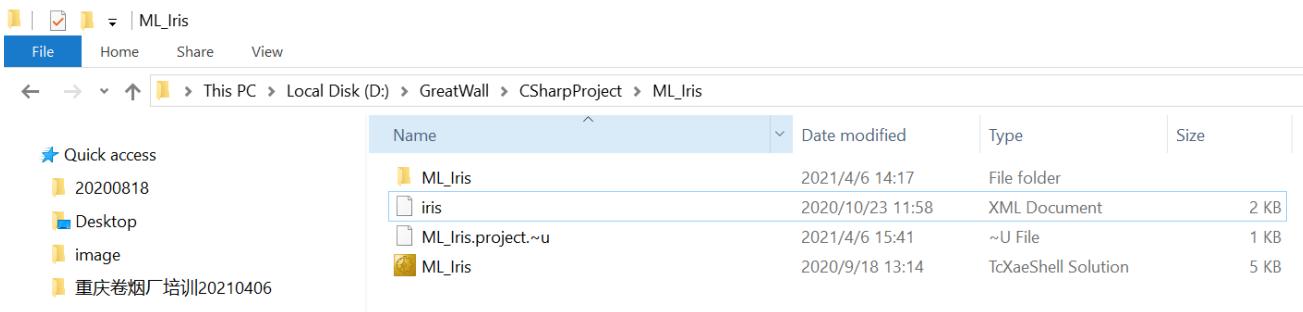
### 2、onnx文件转换xml文件

倍福自己的转换工具：ModelManager



### 3、xml模型文件拷贝

xml模型文件拷贝至tc3工程下，方便日后使用。xml模型文件在路径（根据自己实际情况，也可用2中ModelManager按钮'Open converted files path'） D:\TwinCAT\TC3\Functions\TF38xx-Machine-Learning\ConvertToolFiles



## 六、TwinCat调用运行

**注意：**

我们教程有API方式以及TcCom方式，本例程采用api方式

### 1、导入Tc3\_MLL库文件

### 2、编写程序

```

//定义区
PROGRAM MAIN
VAR
    fbMllPrediction:FB_MllPrediction;
    maxConcurrency:udint;
    nState: INT;
    bLoadConfig: BOOL;
    hr: BOOL;
    idim: UDINT:=4;
    odim: UDINT:=3;
    modelName: T_MaxString:='D:/GreatWall/CSharpProject/ML_Iris/iris.xml';

    bPreferred: BOOL;
    bSupported: BOOL;
    nnInput: ARRAY[1..4] OF REAL:=[5.5,2.3,4.0,1.3];
    nnOutput: ARRAY[1..3] OF REAL;
    nCurrentEngineID: UINT;
END_VAR

//正文区
//fbMllPrediction.stPredictionParameter.MlModelFilepath:='D:/GreatWall/CSharpProject/ML_Iris/iris.xml';

//idim:=USINT_TO_UDINT(SIZEOF(nnInput));
//odim:=USINT_TO_UDINT(SIZEOF(nnOutput));
CASE nState OF
    0:
        //手动置位bLoadConfig
        IF bLoadConfig THEN
            bLoadConfig:=FALSE;
            nState:=10;
        END_IF
    10:
        //xml模型读取

fbMllPrediction.stPredictionParameter.MlModelFilepath:='D:/GreatWall/CSharpProject/ML_Iris/iris.xml';
        IF fbMllPrediction.Configure() THEN
            // hr:=fbMllPrediction.GetInputDim(idim);

            //FAILED
            // hr:=fbMllPrediction.GetOutputDim(odim);
            //
            // hr:=fbMllPrediction.GetModelName(modelName);
            //
            // hr:=fbMllPrediction.GetMaxConcurrency(maxConcurrency);
            //
            //
            hr:=fbMllPrediction.CheckPreferredIODataTypes(ETcMllDataType.E_MLLDT_FP32_REAL,ETcMllDataType.E_MLLDT_FP32_REAL,bPreferred);
            //

```

```

//  

hr:=fbM11Prediction.CheckSupportedIOTypes(ETcM11DataType.E_MLLDT_FP32_REAL,ETcM11DataType.E_  

MLLDT_FP32_REAL,bSupported);  

//  

IF fbM11Prediction.bError THEN  

    nState:=999;  

ELSE  

    nState:=20;  

END_IF  

END_IF  

20:  

//根据输入，计算输出  

fbM11Prediction.Predict(  

    pDataInp:=ADR(nnInput),  

    nDataInpDim:=idim,  

    fmtDataInpType:=ETcM11DataType.E_MLLDT_FP32_REAL ,  

    pDataOut:=ADR(nnOutput) ,  

    nDataOutDim:=odim ,  

    fmtDataOutType:=ETcM11DataType.E_MLLDT_FP32_REAL ,  

    nEngineId:=nCurrentEngineID ,  

    nConcurrencyId:=0 );  

IF fbM11Prediction.bError THEN  

    nState:=999;  

ELSIF bLoadConfig THEN  

    bLoadConfig:=FALSE;  

    nState:=10;  

END_IF  

999:  

;  

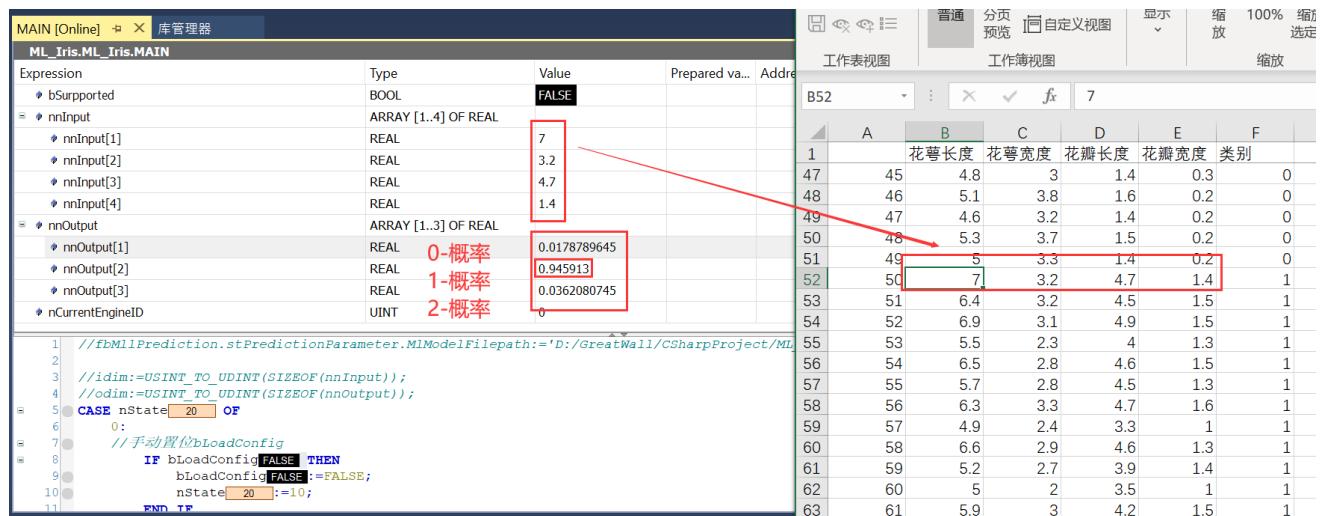
END_CASE

```

### 3、运行测试

nnInput为输入特征值花萼长度，花萼宽度，花瓣长度，花瓣宽度；

nnOutput为输出标签，0-狗尾草鸢尾，1-杂色鸢尾，2-弗吉尼亚鸢尾



The screenshot shows the SIMATIC Manager interface with two main windows. On the left, the 'MAIN [Online]' window displays variable assignments:

Expression	Type	Value	Prepared va...	Address
bSupported	BOOL	FALSE		
= nnInput	ARRAY [1..4] OF REAL			
nnInput[1]	REAL	7		
nnInput[2]	REAL	3.2		
nnInput[3]	REAL	4.7		
nnInput[4]	REAL	1.4		
= nnOutput	ARRAY [1..3] OF REAL			
nnOutput[1]	REAL	0-概率 0.0178789645		
nnOutput[2]	REAL	1-概率 0.945913		
nnOutput[3]	REAL	2-概率 0.0362080745		
nCurrentEngineID	UINT	0		

On the right, a spreadsheet table titled 'B52' shows the Iris dataset:

	A	B	C	D	E	F
		花萼长度	花萼宽度	花瓣长度	花瓣宽度	类别
1						
47	45	4.8	3	1.4	0.3	0
48	46	5.1	3.8	1.6	0.2	0
49	47	4.6	3.2	1.4	0.2	0
50	48	5.3	3.7	1.5	0.2	0
51	49	5	3.3	1.4	0.2	0
52	50	7	3.2	4.7	1.4	1
53	51	6.4	3.2	4.5	1.5	1
54	52	6.9	3.1	4.9	1.5	1
55	53	5.5	2.3	4	1.3	1
56	54	6.5	2.8	4.6	1.5	1
57	55	5.7	2.8	4.5	1.3	1
58	56	6.3	3.3	4.7	1.6	1
59	57	4.9	2.4	3.3	1	1
60	58	6.6	2.9	4.6	1.3	1
61	59	5.2	2.7	3.9	1.4	1
62	60	5	2	3.5	1	1
63	61	5.9	3	4.2	1.5	1

MAIN [Online] 库管理器

**ML\_Iris.ML\_Iris.MAIN**

Expression	Type	Value	Prepared va...	Address
bSupported	BOOL	FALSE		
nnInput	ARRAY [1..4] OF REAL			
nnInput[1]	REAL	6.3		
nnInput[2]	REAL	3.3		
nnInput[3]	REAL	6		
nnInput[4]	REAL	2.5		
nnOutput	ARRAY [1..3] OF REAL			
nnOutput[1]	REAL	0-概率 0.0002929907		
nnOutput[2]	REAL	1-概率 0.16591306		
nnOutput[3]	REAL	0.833793938		
nCurrentEngineID	UINT	2-概率 0		

```

1 //fbMlPrediction.stPredictionParameter.MlModelFilepath='D:/GreatWall/CSharpProject/ML...
2
3 //idim:=USINT_TO_UDINT(SIZEOF(nnInput));
4 //odim:=USINT_TO_UDINT(SIZEOF(nnOutput));
5 CASE nState:=20 OF
6   0:
7     //手动置位bLoadConfig
8     IF bLoadConfig=FALSE THEN
9       bLoadConfig:=FALSE;
10      nState:=10;
11 END_IF

```

Error List

工作表视图 工作簿视图 普通 分页 预览 自定义视图 显示 缩放 100% 组选

B102 : fx 6.3

A	B	C	D	E	F
1	花萼长度	花萼宽度	花瓣长度	花瓣宽度	类别
92	90	5.5	2.6	4.4	1
93	91	6.1	3	4.6	1
94	92	5.8	2.6	4	1
95	93	5	2.3	3.3	1
96	94	5.6	2.7	4.2	1
97	95	5.7	3	4.2	1
98	96	5.7	2.9	4.2	1
99	97	6.2	2.9	4.3	1
100	98	5.1	2.5	3	1
101	99	5.7	2.8	4.1	1
102	100	6.3	3.3	6	2
103	101	5.8	2.7	5.1	1
104	102	7.1	3	5.9	2
105	103	6.3	2.9	5.6	2
106	104	6.5	3	5.8	2
107	105	7.6	3	6.6	2
108	106	4.9	2.5	4.5	1
109	107	7.3	2.9	6.3	1