

# 在java中调用训练好的TensorFlow模型

当我们训练好TensorFlow模型以后，我们往往都是需要投入实际使用的，在实际使用的时候，我们不可能先训练，后处理，因为训练的代价实在是太大了。本文主要讲解如何将训练好的tensorflow模型保存成.pb格式的文档，并在java项目中运用。

- 保存模型
- 在java中调用

先来解决第一个问题，如何保存为pb格式，其实这是非常简单的，只需要3行代码即可。

```
builder = tf.saved_model.builder.SavedModelBuilder('./model2')
# SavedModelBuilder里面放的是你想要保存的路径，比如我的路径是根目录下的model2文件
builder.add_meta_graph_and_variables(session, ["mytag"])
#第二步必需要有，它是给你的模型贴上一个标签，这样再次调用的时候就可以根据标签来找。我给
它起的标签名是"mytag"，你也可以起别的名字，不过你需要记住你起的名字是什么。
builder.save()
#第3步是保存操作
```

其实第一个问题还没有解决，如果你直接这样保存的话，你在调用的时候可能就找不到输入和输出了。所以你需要在你的代码里给你的输入和输出变量起个名字，这样去java里面，你可以根据这个名字来获得你的输入输出变量。

如果你没有理解我这段话，你就看下面这个例子。

```
X_holder = tf.placeholder(tf.int32,[None,None],name='input_x') # 训练集
predict_Y = tf.nn.softmax(softmax_before,name='predict') # softmax() 计算
概率
#就拿我的案例来说，我的输入是一个二维矩阵，我给它起名为"input_x"，这样到了java中我就
可以根据"input_x"来得到x_hodler
#同理，我的输出是一个softmax(),所以我给它起名为"predict"。
#其他变量我们是不用考虑的，因为我们训练模型的目的就是给输入，得到输出。
```

好了，现在第一个问题解决了。现在来解决如何在java中调用的问题。

1. 先在maven的pom.xml中引入tensorflow的包。我的包是这样的

```
<dependency>
    <groupId>org.tensorflow</groupId>
    <artifactId>libtensorflow</artifactId>
    <version>1.12.0</version>
</dependency>
<dependency>
    <groupId>org.tensorflow</groupId>
    <artifactId>proto</artifactId>
    <version>1.12.0</version>
</dependency>
```

```
<dependency>
    <groupId>org.tensorflow</groupId>
    <artifactId>libtensorflow_jni</artifactId>
    <version>1.12.0</version>
</dependency>
```

千万不要直接复制我的代码，因为你训练时使用的tensorflow版本不一定是1.12.0. 所以你需要去python里看看你的TensorFlow版本是多少。方法如下：

```
import tensorflow as tf
tf.__version__
```

得到你的版块以后，将version换一下就行。

接下来你就可以直接复制代码了。

```
import org.tensorflow.*;
SavedModelBundle b = SavedModelBundle.load("./src/main/resources/model2", "mytag");
//.load首先需要的是你打包好的.pb文件所在的目录，其次是刚刚你定义的标签名称
Session tfSession = b.session();
Operation operationPredict = b.graph().operation("predict"); // 要执行的op,根据名字找到输出
Output output = new Output(operationPredict, 0);
Tensor input_X = Tensor.create(input);
//这里的input我没有给出定义，因为这取决于你的模型，这里它是一个二维数组，因为我们模型输入就是一个二维数据，我们需要将二维数组转化为tensor变量
Tensor out= tfSession.runner().feed("input_x",input_X).fetch(output).run().get(0); //输入
//后面的代码不一定一样，取决于你的训练模型本身
System.out.println(out);
float [][] ans = new float[1][10];
out.copyTo(ans); //将tensor里面的数据copy给一个数组
```

算了，我还是把我的全部代码贴出了吧,这样看起来比较直观一些。

```
package com.example.demo.tf;

/**
 * @ClassName Read
 * @Description TODO
 * @Author ydc
 * @Date 2019/2/12 8:21
 * @Version 1.0
 */
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
```

```

import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.math.*;
import java.util.Random;
import org.tensorflow.*;
public class Read {

    private static final Integer ONE = 1;

    public static void main(String[] args) {
        Map<String, Integer> map = new HashMap<String, Integer>();
        Map<Integer,String> mp = new HashMap<>();
        mp.put(0,"体育");
        mp.put(1,"娱乐");
        mp.put(2,"家居");
        mp.put(3,"房产");
        mp.put(4,"教育");
        mp.put(5,"时尚");
        mp.put(6,"时政");
        mp.put(7,"游戏");
        mp.put(8,"科技");
        mp.put(9,"财经");
        try {
            BufferedReader br = new BufferedReader(new
InputStreamReader(new FileInputStream(new File("./src/main/resources/dat
a/vocab.txt")),
                "UTF-8"));
            String lineTxt = null;
            int idx =0 ;
            while ((lineTxt = br.readLine()) != null) {
                map.put(lineTxt,idx);
                idx++;
            }
            br.close();
        } catch (Exception e) {
            System.err.println("read errors :" + e);
        }
        int input [][] =new int[1][600];
        int max=1000;
        int min=1;
        Random random = new Random();
        for(int i=0;i<1;i++){
            for(int j=0;j<600;j++){
                // input[i][j]=random.nextInt(max)%(max-min+1) + min;
                input[i][j]=0;
            }
        }

        try {

```

```

        BufferedReader br = new BufferedReader(new
InputStreamReader(new FileInputStream(new File("./src/main/resources/dat
a/test.txt")),
            "utf-8"));
        String lineTxt = null;
        int idx = 0 ;
        while ((lineTxt = br.readLine()) != null) {
            int sz =lineTxt.length();
            System.out.println(lineTxt);
            for(int k=0;k<1;k++) {
                for (int i = 0; i < sz; i++) {
                    String tmp = String.valueOf(lineTxt.charAt(i));
                    //System.out.print(tmp+" ");
                    if(map.get(tmp)==null){
                        System.out.println(tmp);
                        continue;
                    }
                    input[k][i] = map.get(tmp);
                }
            }
        }
        br.close();
    } catch (Exception e) {
        System.err.println("read errors :" + e);
    }
    for(int i=0;i<600;i++){
        System.out.print(input[0][i]+" ");
        if(i%100==0){
            System.out.println();
        }
    }

    SavedModelBundle b = SavedModelBundle.load("./src/main/resources/
model2", "mytag");
    Session tfSession = b.session();
    Operation operationPredict = b.graph().operation("predict");    //
要执行的op
    Output output = new Output(operationPredict, 0);
    Tensor input_X = Tensor.create(input);
    Tensor out= tfSession.runner().feed("input_x",input_X).fetch(outp
ut).run().get(0);
    System.out.println(out);
    float [][] ans = new float[1][10];
    out.copyTo(ans);
    float M=0;
    int index1=0;
    index1 =getMax(ans[0]);
    System.out.println(index1);
    System.out.println("-----");
    System.out.println(mp.get(index1));

    //System.out.println(mp.get(getMax(ans[1])));
}

```

```
public static int getMax(float[] a){  
    float M=0;  
    int index2=0;  
    for(int i=0;i<10;i++){  
        if(a[i]>M){  
            M=a[i];  
            index2=i;  
        }  
    }  
    return index2;  
}
```