机器学习项目接口

若借助本文所述系统，将一个机器学习项目分布式部署在边缘端和云端

需要使用系统的开发者，对机器学习项目进行合理封装，提供训练、推断所需必要的接口。

本节将阐述本系统针对图像分类智能应用所给出的通用接口的设计，并以基于tensorflow框架的人脸表情识别的机器学习模型为例，给出实现本系统接口、完成机器学习模型部署的实例。

**技术背景**

如今，基于tensorflow框架的机器学习项目的占据绝大部分，因此本系统的接口也主要针对tensorflow框架的标准进行适配。

**Tensorflow模型训练、保存和读取**

Tensorflow通过tf.Session类实现神经网络模型的训练

TensorFlow通过tf.train.Saver类实现神经网络模型的保存和提取[]。

**模型的保存**

Saver对象的创建依赖于一个Session实例，在训练过程中、每一轮训练结束后，都可以通过调用其save方法将模型当前状态保存到指定路径；实践中，常常配置间隔固定轮次保存一次模型，以及时观察和备份训练成果，降低风险。在该路径下，将会产生四种文件，如表[]所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **后缀名** | **文件类型** | **文件命名** | **说明** |
| 无 | 文本文件 | checkpoint | 目录下唯一，用于组织训练过程中保存的全部模型信息列表和最新保存的模型信息。 |
| .meta | 二进制 | {模型名称}\_{起始训练轮数}-{保存时训练轮数} | 保存对应模型的TensorFlow计算图的结构信息 |
| .index | 二进制 | 保存对应模型神经网络中每个变量的索引 |
| .data | 二进制 | 保存对应模型神经网络中每个变量的取值 |

在训练、模型保存的过程中Tensorflow框架以checkpoint文件维护model\_checkpoint\_path和all\_model\_checkpoint\_path两个字段，前者表示最新模型绝对路径后者表示所有保存的模型的绝对路径列表。值得说明的是，checkpoint文件内容在结束训练之后固定不随项目、文件环境的变化而更改，在网络传输后失效。因此，本文不通过checkpoint文件来获取模型。

**模型的读取**

Tensorflow提供两种模型的读取方式

对于模型的应用，最优方式是

而如果想在读取模型后，在其基础上进行增量训练，则必须采用

首先，需要调用前向传播模块重新构建与原模型结构相同的输入、输出占位符和网络结构，以此创建一个空的Saver实例，再通过其restore方法，传入模型路径、加载指定模型。

对于本文所构建的集合应用与持续优化的系统，将分别在云端的增量训练、边缘端的训练分别采取两种读取方式。

**基于Tensorflow框架的机器学习项目接口设计与实现**

实时推断接口

请求参数：

|  |  |  |
| --- | --- | --- |
| **形参** | **数据类型** | **说明** |
| testPic | str | 待推断图片在本地的完整路径 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **形参** | **数据类型** | **说明** |
| label | Int | 推断分类结果编号 |

实现逻辑示例：

视觉类智能应用，往往对实时性有着很高要求，为了提高模型响应水平，最大化利用计算资源，边缘端程序提供实时推断服务期间，在没有新模型被下发的情形下，推断会话持续维护在内存以内，不应反复加载。

具体流程图如下：

由于参数

模型的提取

模型的加载将会耗费较多时间，为了保证应用的实时性，加载好的模型应常驻内存，以在获取用户源数据后能够迅速响应。只有在获取更换模型的命令时，才会重新加载模型参数。所以，循环将在Session内部。使用yield方法，使返回结果的同时，模型保留在内存中。

python 指定函数输入输出类型

def fun(i: int) -> bool: # 指定输入为int，输出为bool类型

print(i)

return bool(i)

最新模型的自动加载

train.get\_checkpoint\_state(MODEL\_SAVE\_PATH)

指定模型的加载